

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。

アプリケーション・ノート

保守/廃止

μPD78014 サブシリーズ

8ビット・シングルチップ・マイクロコンピュータ

電子手帳編

μPD78014

μPD78P014

〔メモ〕

目次要約

第 1 章	機能概要	... 17
第 2 章	ハードウェア	... 19
第 3 章	機能説明	... 25
第 4 章	プログラム説明	... 69
付 録	SPDチャートの説明	... 413

CMOSデバイスの一般的注意事項

静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレイやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

本製品のうち、外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当するものについては、日本国外に輸出する際に、同法に基づき日本国政府の輸出許可が必要です。

- 本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。
 - 文書による当社の承諾なしに本資料の転載複製を禁じます。
 - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
 - 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。
 - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
 - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
 - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
 - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート／データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

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

はじめに

対象者 このアプリケーション・ノートは、μPD78014サブシリーズ製品の機能を理解し、μPD78014サブシリーズ製品を用いたアプリケーション・システムを設計するユーザのエンジニアを対象としています。

対象製品は次のとおりです。

μPD78014サブシリーズ : μPD78014, 78P014

目的 このアプリケーション・ノートは、μPD78014サブシリーズを用いた例として、電子手帳の機能と構成をユーザに理解していただくことを目的とします。

なお、記載しているプログラムおよびハードウェアの構成は例示的に示したものであり、量産設計を対象とするものではありません。

構成 このアプリケーション・ノートは、大きく分けて次の内容で構成しています。

機能概要
ハードウェア
機能説明
プログラム説明
SPDチャートの説明

読み方 このアプリケーション・ノートの読者には、電気、論理回路、マイクロコンピュータの一般知識を必要とします。

このアプリケーション・ノートでは、使用例として、μPD78014サブシリーズのうち、μPD78014を使って説明しています。

凡例 データ表記の重み : 左が上位桁，右が下位桁
アクティブ・ロウの表記 : $\overline{\text{xxx}}$ (端子，信号名称に上線)
注 : 本文中につけた注の説明
注意 : 気を付けて読んでいただきたい内容
備考 : 本文の補足説明
数の表記 : 2進数..... xxx または xxxxB
 : 10進数..... xxx
 : 16進数..... xxxH

品質水準

オーダ名称	パッケージ	品質水準
μ PD78014CW- x x x	64ピン・プラスチック・シュリンクDIP (19.05 mm (750))	標準 (一般電子機器用)
μ PD78014GC- x x x-AB8	64ピン・プラスチックQFP (14 x 14)	〃
μ PD78P014CW	64ピン・プラスチック・シュリンクDIP (19.05 mm (750))	〃
μ PD78P014DW	64ピン・セラミック・シュリンクDIP (窓付き) (19.05 mm (750))	適用外 (機能評価用)
μ PD78P014GC-AB8	64ピン・プラスチックQFP (14 x 14)	標準 (一般電子機器用)

注意 μPD78P014DWは、お客様の装置の量産製品に使用されることを意図した信頼性を保持しておりません。実験または機能評価用のみご使用ください。

備考 x x xはROMコード番号です。

品質水準とその応用分野の詳細については当社発行の資料「NEC半導体デバイスの品質水準」(資料番号 C11531J)をご覧ください。

応用分野 電話、VTR、オーディオ、カメラ、家電製品など

関連資料

資料名	資料番号	
	和文	英文
μ PD78014, 78014Yサブシリーズ ユーザーズ・マニュアル	U10085J	U10085E
μ PD78011B, 78012B, 78013, 78014 データ・シート	IC-8201	IC-3179
μ PD78P014 データ・シート	IC-8111	IC-3098
μ PD78014, 78014Yサブシリーズ 特殊機能レジスタ活用表	IEM-5527	-
78K/0シリーズ ユーザーズ・マニュアル 命令編	U12326J	U12326E
78K/0シリーズ アプリケーション・ノート 基礎編 ()	U12704J	IEA-1288
μ PD78014サブシリーズ アプリケーション・ノート 電子手帳編	このマニュアル	IEA-1301

注意 上記関連資料は予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。

目 次

- 第1章 機能概要 ... 17**
 - 1.1 名刺（個人／会社）機能 ... 18
 - 1.2 カレンダー機能 ... 18
 - 1.3 メモ帳機能 ... 18
 - 1.4 世界時計機能 ... 18
 - 1.5 通信機能 ... 18

- 第2章 ハードウェア ... 19**
 - 2.1 ハードウェア構成 ... 19
 - 2.2 メモリ・マップ ... 21
 - 2.3 LCDモジュールの説明 ... 22

- 第3章 機能説明 ... 25**
 - 3.1 キー入力 ... 26
 - 3.2 メニュー・モード ... 27
 - 3.3 名刺（個人）機能 ... 28
 - 3.3.1 登録モード ... 29
 - 3.3.2 データ選択モード ... 30
 - 3.3.3 修正モード ... 32
 - 3.3.4 検索モード ... 33
 - 3.3.5 削除モード ... 34
 - 3.4 名刺（会社）機能 ... 35
 - 3.4.1 登録モード ... 36
 - 3.4.2 データ選択モード ... 37
 - 3.4.3 修正モード ... 39
 - 3.4.4 検索モード ... 40
 - 3.4.5 削除モード ... 41
 - 3.5 カレンダー機能 ... 42
 - 3.5.1 現時刻のカレンダー表示 ... 43
 - 3.5.2 カレンダー表示の月設定 ... 47
 - 3.5.3 休日の設定 ... 49
 - 3.6 メモ帳機能 ... 52
 - 3.6.1 登録モード ... 53
 - 3.6.2 データ選択モード ... 55
 - 3.6.3 修正モード ... 58
 - 3.6.4 検索モード ... 59
 - 3.6.5 削除モード ... 60
 - 3.7 世界時計機能 ... 61
 - 3.7.1 時計表示モード ... 62
 - 3.7.2 時計表示の都市選択モード ... 63

- 3.8 初期設定 ... 64
 - 3.8.1 日付設定モード ... 65
 - 3.8.2 時間設定モード ... 66
- 3.9 通信機能 ... 67
 - 3.9.1 電子手帳とPC-9801シリーズの通信方法 ... 68

第4章 プログラム説明 ... 69

- 4.1 モジュール説明 ... 69
 - 4.1.1 モジュール構成 ... 69
 - 4.1.2 ジェネラルSPDチャート ... 70
- 4.2 RAM説明 ... 72
- 4.3 メイン ... 75
 - 4.3.1 イニシャライズ ... 75
 - 4.3.2 メニュー ... 75
 - 4.3.3 キー入力 ... 76
 - 4.3.4 LCD表示 ... 76
 - 4.3.5 時計カウント ... 77
 - 4.3.6 パワーオフ処理 ... 77
 - 4.3.7 共通処理 ... 77
 - 4.3.8 SPDチャート ... 78
 - 4.3.9 プログラム・リスト ... 94
- 4.4 名刺(個人)モード ... 125
 - 4.4.1 登録データのフォーマット ... 125
 - 4.4.2 モード選択条件 ... 126
 - 4.4.3 登録処理 ... 127
 - 4.4.4 データ選択処理 ... 127
 - 4.4.5 修正処理 ... 128
 - 4.4.6 検索処理 ... 129
 - 4.4.7 削除処理 ... 129
 - 4.4.8 SPDチャート ... 130
 - 4.4.9 プログラム・リスト ... 145
- 4.5 名刺(会社)モード ... 169
 - 4.5.1 登録データのフォーマット ... 169
 - 4.5.2 モード選択条件 ... 170
 - 4.5.3 登録処理 ... 171
 - 4.5.4 データ選択処理 ... 171
 - 4.5.5 修正処理 ... 172
 - 4.5.6 検索処理 ... 173
 - 4.5.7 削除処理 ... 173
 - 4.5.8 SPDチャート ... 174
 - 4.5.9 プログラム・リスト ... 190
- 4.6 カレンダー・モード ... 215
 - 4.6.1 登録データのフォーマット ... 215
 - 4.6.2 モード選択条件 ... 216
 - 4.6.3 カレンダー表示処理 ... 217
 - 4.6.4 スケジュール登録処理 ... 220
 - 4.6.5 スケジュール表示, 修正処理 ... 221

- 4.6.6 スケジュール削除処理 ... 221
- 4.6.7 カレンダー表示月の設定処理 ... 222
- 4.6.8 指定休日の設定処理 ... 222
- 4.6.9 週休の設定処理 ... 222
- 4.6.10 SPDチャート ... 223
- 4.6.11 プログラム・リスト ... 252
- 4.7 メモ帳モード ... 294**
 - 4.7.1 登録データのフォーマット ... 294
 - 4.7.2 モード選択条件 ... 295
 - 4.7.3 登録処理 ... 296
 - 4.7.4 データ選択処理 ... 297
 - 4.7.5 修正処理 ... 298
 - 4.7.6 検索処理 ... 299
 - 4.7.7 削除処理 ... 299
 - 4.7.8 SPDチャート ... 300
 - 4.7.9 プログラム・リスト ... 319
- 4.8 世界時計モード ... 355**
 - 4.8.1 モード選択条件 ... 355
 - 4.8.2 時計表示処理 ... 355
 - 4.8.3 都市選択処理 ... 356
 - 4.8.4 SPDチャート ... 357
 - 4.8.5 プログラム・リスト ... 359
- 4.9 初期設定モード ... 367**
 - 4.9.1 モード選択条件 ... 367
 - 4.9.2 日付の設定 ... 367
 - 4.9.3 時刻の設定 ... 367
 - 4.9.4 SPDチャート ... 368
 - 4.9.5 プログラム・リスト ... 373
- 4.10 通信モード ... 381**
 - 4.10.1 モード選択条件 ... 381
 - 4.10.2 PC - 9801シリーズとのデータ通信 ... 381
 - 4.10.3 電子手帳同士のデータ通信 ... 384
 - 4.10.4 SPDチャート ... 386
 - 4.10.5 プログラム・リスト ... 393
- 4.11 表示データ・テーブル ... 404**
 - 4.11.1 プログラム・リスト ... 405
- 付 録 SPDチャートの説明 ... 413**

図の目次（1/3）

図番号	タイトル, ページ
1 - 1	電子手帳の機能構成 ... 17
2 - 1	電子手帳ハードウェア構成 ... 19
2 - 2	μPD78014メモリ・マップ ... 21
3 - 1	オートパワーOFF ... 26
3 - 2	メニュー表示 ... 27
3 - 3	名刺（個人）モード選択表示 ... 28
3 - 4	名刺（個人）登録モードの表示 ... 29
3 - 5	名刺（個人）先頭文字選択モードの表示 ... 30
3 - 6	名刺（個人）検索データ選択モードの表示 ... 31
3 - 7	名刺（個人）修正モードの表示 ... 32
3 - 8	名刺（個人）検索モードの表示 ... 33
3 - 9	名刺（個人）会社データ一覧の表示 ... 34
3 - 10	名刺（個人）削除モードの表示 ... 34
3 - 11	名刺（会社）モード選択表示 ... 35
3 - 12	名刺（会社）登録モードの表示 ... 36
3 - 13	名刺（会社）先頭文字選択モードの表示 ... 37
3 - 14	名刺（会社）検索データ選択モードの表示 ... 38
3 - 15	名刺（会社）修正モードの表示 ... 39
3 - 16	名刺（会社）検索モードの表示 ... 40
3 - 17	名刺（会社）会社員一覧の表示 ... 41
3 - 18	名刺（会社）削除モードの表示 ... 41
3 - 19	カレンダー・モード選択表示 ... 42
3 - 20	現時刻のカレンダー表示 ... 43
3 - 21	スケジュール・モード選択表示 ... 44
3 - 22	スケジュール登録モードの表示 ... 45
3 - 23	スケジュール表示（修正）モードの表示 ... 46
3 - 24	カレンダー表示の月設定モードの表示 ... 47
3 - 25	月設定時のカレンダー表示 ... 48
3 - 26	休日の設定モード選択表示 ... 49
3 - 27	週休モードの表示 ... 50
3 - 28	日付指定モードの表示 ... 51
3 - 29	メモ帳モード選択表示 ... 52

図の目次 (2/3)

図番号	タイトル, ページ
3 - 30	メモ帳タイトル登録モードの表示 ... 53
3 - 31	メモ帳データ登録モードの表示 ... 54
3 - 32	メモ帳タイトル先頭文字選択モードの表示 ... 55
3 - 33	タイトル(レベル1)選択モードの表示 ... 56
3 - 34	タイトル(レベル2)選択モードの表示 ... 57
3 - 35	メモ帳修正モードの表示 ... 58
3 - 36	メモ帳検索モードの表示 ... 59
3 - 37	メモ帳削除モードの表示 ... 60
3 - 38	世界時計モード選択表示 ... 61
3 - 39	時計表示モードの表示 ... 62
3 - 40	時計表示の都市選択モードの表示 ... 63
3 - 41	初期設定モード選択表示 ... 64
3 - 42	日付設定モードの表示 ... 65
3 - 43	時間設定モードの表示 ... 66
3 - 44	通信モード選択表示 ... 67
4 - 1	電子手帳モジュール構成図 ... 69
4 - 2	LCD反転表示方法 ... 76
4 - 3	名刺(個人)データの登録フォーマット ... 125
4 - 4	名刺(個人)データの検索 ... 127
4 - 5	名刺(個人)の名前データの検索 ... 128
4 - 6	名刺(個人)の会社名データ ... 129
4 - 7	名刺(会社)データの登録フォーマット ... 169
4 - 8	名刺(会社)データの検索 ... 171
4 - 9	名刺(会社)の会社名データの検索 ... 172
4 - 10	名刺(個人)の会社名データ ... 173
4 - 11	スケジュール・データの登録フォーマット ... 215
4 - 12	指定休日の登録フォーマット ... 216
4 - 13	スケジュール・データのフォーマット ... 219
4 - 14	週休データ ... 219
4 - 15	タイトル(レベル1)の登録フォーマット ... 294
4 - 16	メモ帳データの登録フォーマット ... 294
4 - 17	メモ帳データの検索 ... 297
4 - 18	タイトル(レベル2)の検索 ... 298

図の目次 (3/3)

図番号	タイトル, ページ
4 - 19	時差データの構成 ... 355
4 - 20	PC - 9801シリーズの通信フォーマット ... 382
4 - 21	電子手帳同士の通信フォーマット ... 384
4 - 22	LCD表示データ形式 ... 404

表の目次（1/2）

表番号	タイトル, ページ
2 - 1	μPD78014端子表 ... 20
2 - 2	EG2402S - AR端子表 ... 22
2 - 3	EG2402S - ARの使用コマンド一覧 ... 23
3 - 1	電子手帳の使用キー一覧 ... 26
3 - 2	メニュー・モードの有効キー ... 27
3 - 3	名刺（個人）モード選択の有効キー ... 28
3 - 4	名刺（個人）登録モードの有効キー ... 29
3 - 5	名刺（個人）先頭文字選択モードの有効キー ... 30
3 - 6	名刺（個人）検索データ選択モードの有効キー ... 31
3 - 7	名刺（個人）修正モードの有効キー ... 32
3 - 8	名刺（個人）検索モードの有効キー ... 33
3 - 9	名刺（個人）会社データ一覧の有効キー ... 34
3 - 10	名刺（個人）削除モードの有効キー ... 34
3 - 11	名刺（会社）モード選択の有効キー ... 35
3 - 12	名刺（会社）登録モードの有効キー ... 36
3 - 13	名刺（会社）先頭文字選択モードの有効キー ... 37
3 - 14	名刺（会社）検索データ選択モードの有効キー ... 38
3 - 15	名刺（会社）修正モードの有効キー ... 39
3 - 16	名刺（会社）検索モードの有効キー ... 40
3 - 17	名刺（会社）会社員一覧の有効キー ... 41
3 - 18	名刺（会社）削除モードの有効キー ... 41
3 - 19	カレンダー・モード選択の有効キー ... 42
3 - 20	現時刻のカレンダー表示の有効キー ... 43
3 - 21	スケジュール・モード選択の有効キー ... 44
3 - 22	スケジュール登録モードの有効キー ... 45
3 - 23	スケジュール表示（修正）モードの有効キー ... 46
3 - 24	カレンダー表示の月設定モードの有効キー ... 47
3 - 25	月設定時のカレンダー表示の有効キー ... 48
3 - 26	休日の設定モード選択の有効キー ... 49
3 - 27	週休モードの有効キー ... 50
3 - 28	日付指定モードの有効キー ... 51
3 - 29	メモ帳モード選択の有効キー ... 52
3 - 30	メモ帳タイトル登録モードの有効キー ... 53
3 - 31	メモ帳データ登録モードの有効キー ... 54

表の目次 (2/2)

表番号	タイトル, ページ
3 - 32	メモ帳タイトル先頭文字選択モードの有効キー ... 55
3 - 33	タイトル(レベル1)選択モードの有効キー ... 56
3 - 34	タイトル(レベル2)選択モードの有効キー ... 57
3 - 35	メモ帳修正モードの有効キー ... 58
3 - 36	メモ帳検索モードの有効キー ... 59
3 - 37	メモ帳削除モードの有効キー ... 60
3 - 38	世界時計モード選択の有効キー ... 61
3 - 39	時計表示モードの有効キー ... 62
3 - 40	時計表示の都市選択モードの有効キー ... 63
3 - 41	初期設定モード選択の有効キー ... 64
3 - 42	日付設定モードの有効キー ... 65
3 - 43	時間設定モードの有効キー ... 66
3 - 44	通信モード選択の有効キー ... 67
4 - 1	μPD78014電子手帳RAM一覧表 ... 72
4 - 2	μPD78014電子手帳フラグ一覧表 ... 74
4 - 3	メニュー・モードのMODE0-MODE4の状態 ... 75
4 - 4	パワーオフ・モードのMODE0-MODE4の状態 ... 77
4 - 5	名刺(個人)モードのMODE0-MODE4の状態 ... 126
4 - 6	名刺(会社)モードのMODE0-MODE4の状態 ... 170
4 - 7	カレンダー・モードのMODE0-MODE4の状態 ... 216
4 - 8	メモ帳モードのMODE0-MODE4の状態 ... 295
4 - 9	世界時計モードのMODE0-MODE4の状態 ... 355
4 - 10	世界時計データ ... 356
4 - 11	初期設定モードのMODE0-MODE4の状態 ... 367
4 - 12	送信時のデータ量の違い ... 381
4 - 13	通信モードのTRMOD1-TRMOD3の状態 ... 381
付 - 1	SPD記号とフロー・チャートの対比 ... 413

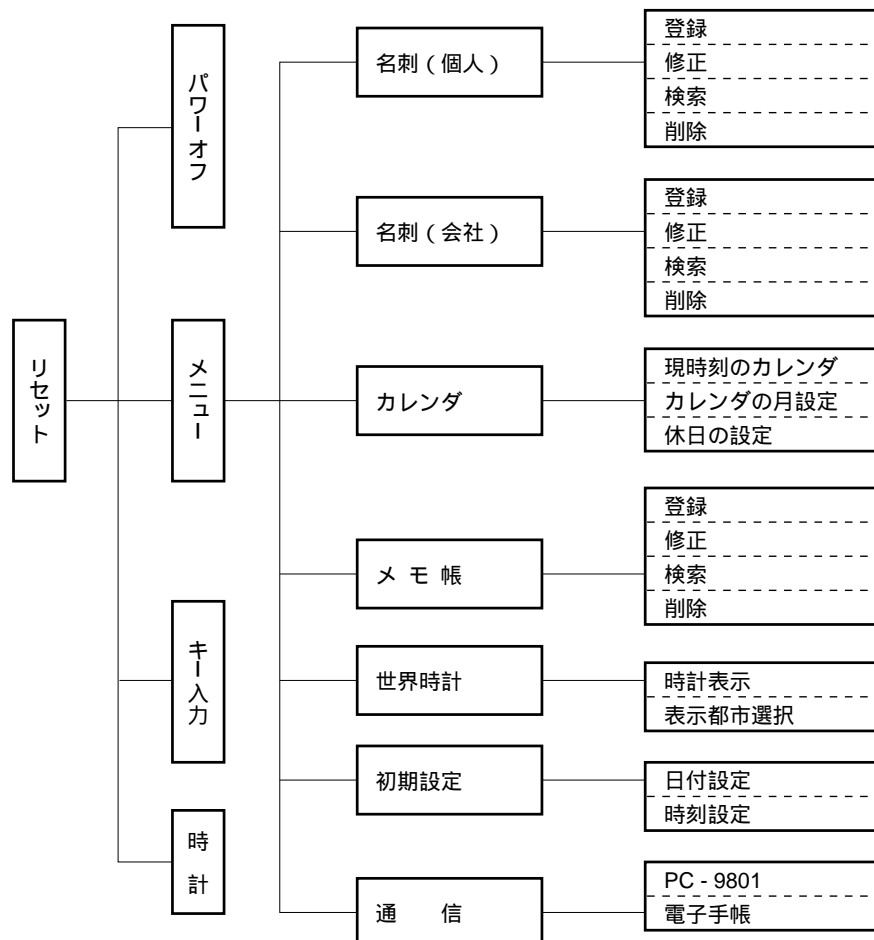
第 1 章 機能概要

このアプリケーション・ノートでは、8ビット・シングルチップマイコンμPD78014を使用して、次の機能を持った電子手帳の応用例を紹介しています。

- 名刺（個人 / 会社）
- カレンダー
- メモ帳
- 世界時計
- 通信

これらの機能は、8 × 8 キー・マトリクスにより操作し、データの登録、修正、検索、削除や各モードの表示をします。ここでは、英数字（A-Z, 0-9）と記号の表示が可能な、30桁 × 8 行のLCDモジュールを使用して表示を行っています。

図 1 - 1 電子手帳の機能構成



1.1 名刺（個人／会社）機能

名刺機能では、個人別と会社別の2種類の入力モードがあります。それぞれのモードで名前、住所、電話番号などを入力します。また、個人、会社間でのデータ検索を行うことができますので、個人モードでの会社データの表示や、会社モードでの会社員一覧表示ができます。

1.2 カレンダー機能

カレンダー機能では、1901年から2099年までのカレンダー表示を行うことができます。そして、カレンダー表示には、休日、スケジュールを示す機能があります。また、スケジュール設定は、日付、時間、用件、アラームON/OFF制御を入力することができます。

1.3 メモ帳機能

メモ帳機能には、タイトルを除き、自由にデータを入力することができます。

1.4 世界時計機能

世界時計機能では、現在時刻、および海外の都市1箇所を表示することができます。都市は、キー操作によりメニュー形式で選択をします。

1.5 通信機能

通信機能では、電子手帳同士の場合と、PC - 9801シリーズとの場合の2種類のモードがあります。電子手帳同士の通信では同期式通信を行い、PC - 9801シリーズとの通信ではRS - 232Cを使用した非同期式通信を行っています。

第2章 ハードウェア

2.1 ハードウェア構成

μPD78014を使用したときの電子手帳のハードウェア構成を図2 - 1に示します。データ入力の機能として、8 × 8キー・マトリクスを使用します。また、電子手帳同士や、PC - 9801シリーズと、通信を利用して外部への入出力を行うことができます。データ表示機能としては、LCDモジュールEG2402S - AR (240 × 64ドット：セイコーエプソン(株)製)を使用して行います。EG2402は、8ビット・パラレル・データ制御により、キャラクタ/グラフィック表示を行うことができます。

電子手帳のデータ・メモリとしては、外部メモリにμPD431000を接続して64 Kバイト (16 K × 4バンク構成)を使用します。その他の機能としては、アラーム出力用にブザーがあります。

図2 - 1 電子手帳ハードウェア構成

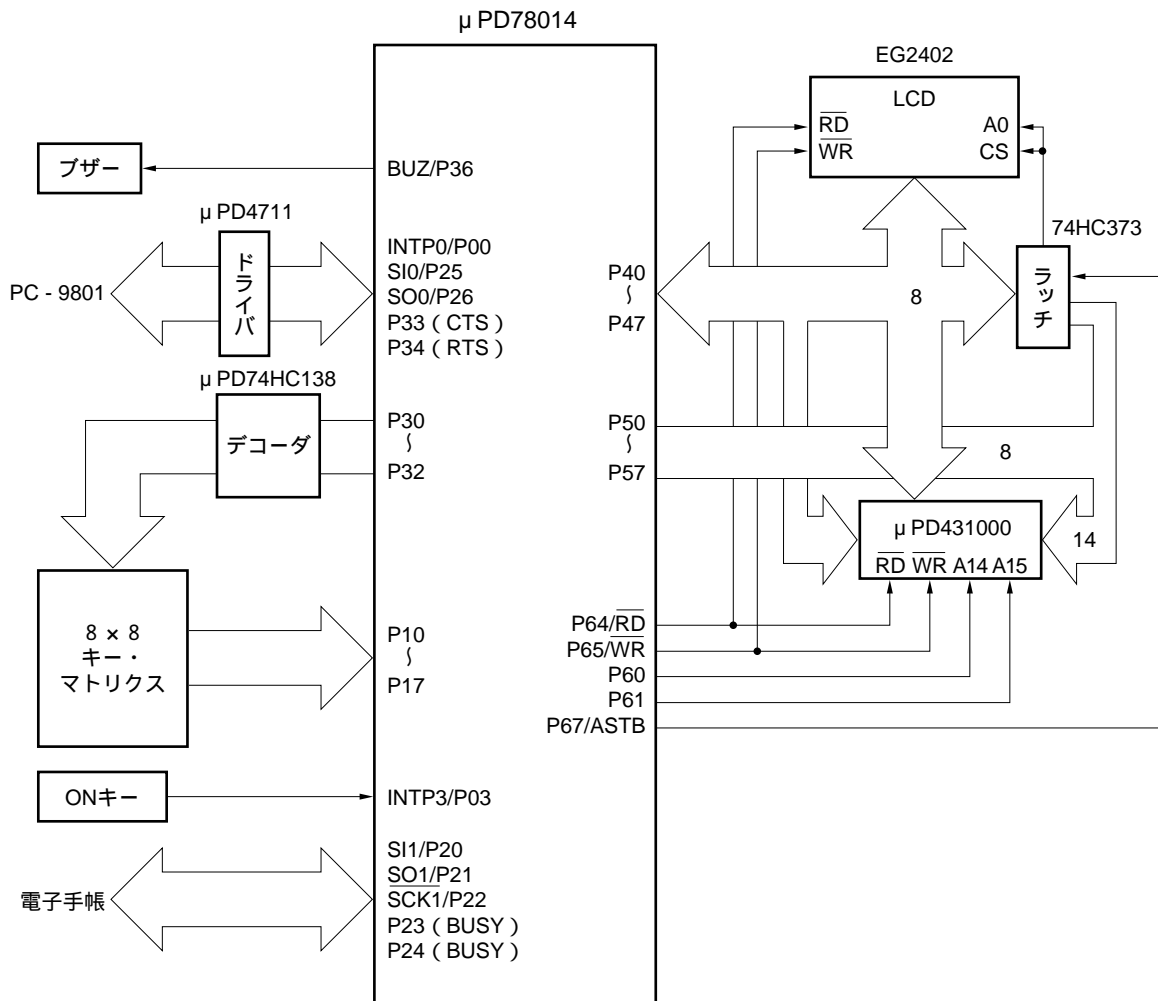


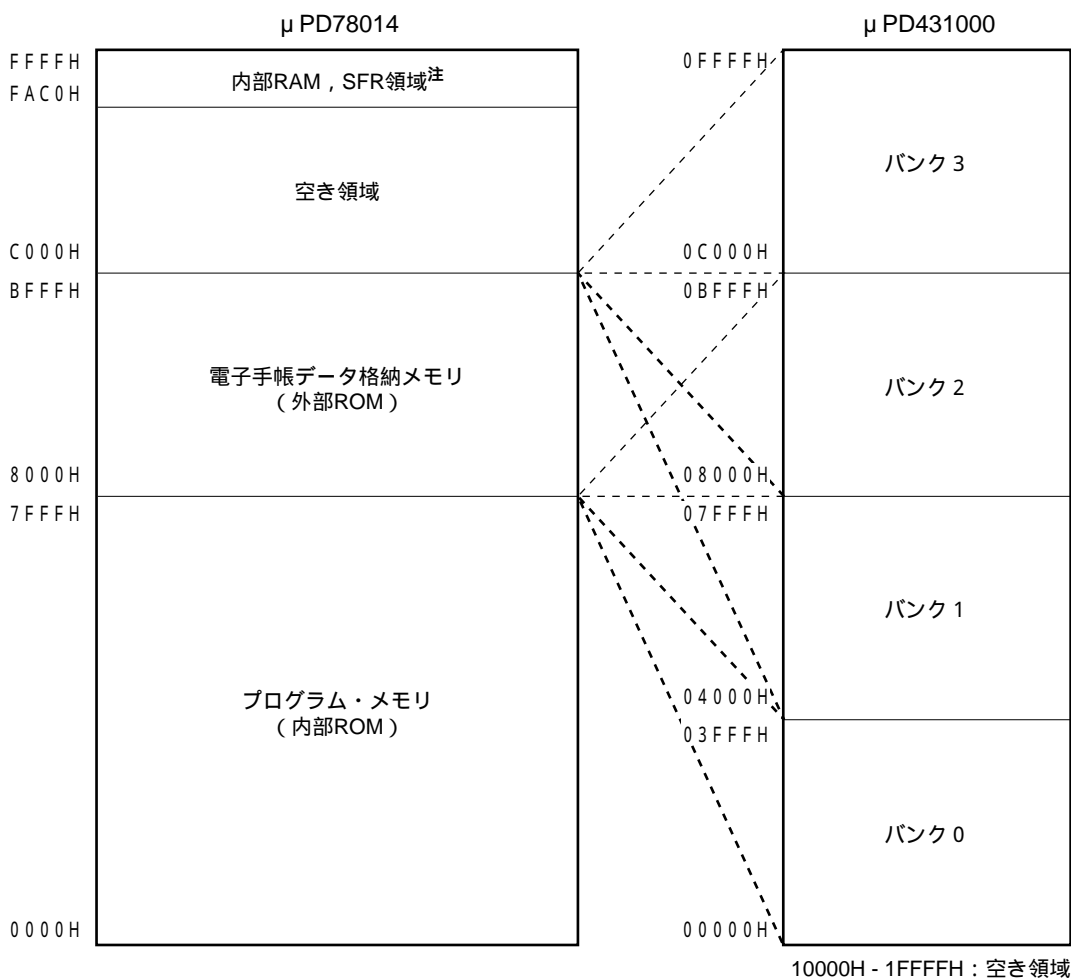
表2-1 μ PD78014端子表

端子番号	I/O	端子名, 用途	アクティブ・レベル	端子番号	I/O	端子名, 用途	アクティブ・レベル	
1	I	P20/SI1, シリアル・データ入力	-	33	O	P56 } アドレス(上位)出力	-	
2	O	P21/SO1, シリアル・データ出力	-	34	O			
3	IO	P22/ $\overline{\text{SCK1}}$, シリアル・クロック入出力	-	35	O	P60 } 外部メモリ・バンク制御	-	
4	I	P23, ビジィ信号入力	L	36	O			
5	O	P24, ビジィ信号出力	L	37	I	P62, 未使用	-	
6	I	P25/SI0, RS-232Cデータ入力	-	38	I	P63, 未使用	-	
7	O	P26/SO0, RS-232Cデータ出力	-	39	O	P64/ $\overline{\text{RD}}$, 外部メモリ $\overline{\text{RD}}$ 信号出力	L	
8	O	P27, μ PD4711スタンバイ信号出力	H	40	O	P65/ $\overline{\text{WR}}$, 外部メモリ $\overline{\text{WR}}$ 信号出力	L	
9	O	P30 } キー・スキャン信号 (74HC138制御信号)	-	41	I	P66, 未使用	-	
10	O			42	O	P67/ASTB, 外部メモリASTB信号	H	
11	O			P32 }	43	I	$\overline{\text{RESET}}$	L
12	I	P33, RS-232C CTS入力	H	44	I	P00/INTP0, RS-232Cスタート検出	L	
13	O	P34, RS-232C RTS出力	H	45	I	P01, 未使用	-	
14	I	P35, 未使用	-	46	I	P02/INTP2, 未使用	-	
15	O	P36/BUZ, ブザー信号出力	-	47	I	P03/INTP3, パワーオン検出	L	
16	I	P37, 未使用	-	48	-	V _{DD}	-	
17	-	V _{SS}	-	49	-	X2, 発振子(8.39 MHz)	-	
18	IO	P40 } アドレス(下位)出力 / データ入出力	-	50	-	X1, 発振子(8.39 MHz)	-	
19	IO			P41	51	-	V _{PP}	-
20	IO			P42	52	-	XT2, 発振子(32.768 kHz)	-
21	IO			P43	53	-	XT1, 発振子(32.768 kHz)	-
22	IO			P44	54	-	AV _{SS} , V _{SS}	-
23	IO			P45	55	I	P10 } キー・リターン信号	L
24	IO			P46	56	I		
25	IO	P47	57	I				
26	O	P50 } アドレス(上位)出力	58	I				
27	O		P51	59	I			
28	O		P52	60	I			
29	O		P53	61	I			
30	O		P54	62	I			
31	O	P55	63	-	AV _{DD} , V _{DD}	-		
32	-	V _{SS}	-	64	-	AV _{REF} , V _{DD}	-	

2.2 メモリ・マップ

μPD78014内部ROM32 Kバイト，内部RAM1 Kバイトのほかに，電子手帳への登録データ格納用に外部RAM64 Kバイトを加えています。外部RAMには，μPD431000を接続し，00000H - 0FFFFHの64 Kバイトを4バンクの構成で使用しています。したがって，1バンクは16 Kバイトとなり，μPD78014の8000H - BFFFFHのアドレス領域にマッピングしています。また，LCDモジュールについては，外部アクセス領域のFFD0H，FFD1Hにマッピングし操作しています。

図2 - 2 μPD78014メモリ・マップ




注 FFD0H，FFD1HにLCDモジュールをマッピング

2.3 LCDモジュールの説明

この電子手帳では、表示用にセイコーエプソン（株）製のLCDモジュールEG2402S - ARを使用しています。EG2402S - ARの特徴を次に示します。

- ・ 240×64ドットのLCDを使用したグラフィック表示と、40×8のキャラクタ表示が可能
- ・ 80系データ・バスのインタフェースによるコントロール（表2 - 2参照）
- ・ JIS, ASCII準拠の160文字キャラクタ・ジェネレータを内蔵
- ・ 画面の重ね合わせによる合成画面が可能（反転表示に使用）
- ・ 5V単一電源
- ・ コントロールLSIにE - 1330を使用

表2 - 2 EG2402S - AR端子表

端子	記号	I/O	名称	機能
1	V _{DD}	I	モジュール電源	+ 5V ± 5%
2	V _{SS}	I	GND	0V
3	NC	-	-	-
4	$\overline{\text{RES}}$	I	リセット	リセット入力
5	$\overline{\text{CS}}$	I	チップ・セレクト	CS = L : アクセス可能 (アドレス・バスのデコードによる信号)
6	A0	I	アドレス	A0 = L : コマンド, H : データ操作 (アドレス・バスの最下位ビットを接続)
7	$\overline{\text{WR}}$	I	ライト	WR = L : 書き込み動作
8	$\overline{\text{RD}}$	I	リード	RD = L : 読み出し操作
9 } 16	D0 } D7	I/O	データ・バス	8ビットのデータ・バス μPD78014のデータ・バスに接続
17	SEL	I	インタフェース選択	SEL = H : 80系インタフェース
18	VR1	I	コントラスト調整用ポ	 VR = 5 - 10 k を接続
19	VR2	O	リユーム接続端子	
20	NC	-	-	-

次に電子手帳の表示に使用するコマンドについて説明します。ここでは、使用するコマンドとパラメータについてのみ説明します。詳細につきましては、EG2402S - AR（セイコーエプソン（株）製）の取扱説明書や、E - 1330の取扱説明書を参照してください。

表2 - 3 EG2402S - ARの使用コマンド一覧

制御内容	コマンド	コードBIN D7 ~ D0	コマンドの説明
動作	SYSTEM SET	01000000	初期化動作，ウインドウ・サイズの設定
	SLEEP IN	01010011	スリープ動作
表示	DISP ON/OFF	0101100D	表示オン/オフ（D = 1/0）の表示，画面点滅指示
	SCROLL	01000100	表示開始アドレス，表示領域の設定
	CSRFORM	01011101	カーソル形状などの設定
	CSRDIR	010011D ₀ D ₁	カーソル移動方向の設定
	OVLAY	01011011	画面重ね合わせモードの指示
	HDOT SCR	01011010	水平方向ドットスクロールのドット数の設定
描画	CSRW	01000110	カーソル・アドレスの設定
メモリ	MWRITE	01000010	表示メモリへの書き込み準備

EG2402S - ARを，30桁 × 8行として使う場合の各コマンドの設定例を次に示します。

SYSTEM SET

C = 01000000B

P1 = 00110000B（内部CG使用）

P2 = 87H（ $240 \div 30 - 1 = 7$ ）（1文字のドット数8ドット）

P3 = 07H（ $64 \div 8 - 1 = 7$ ）（1行のドット数8ドット）

P4 = 29（ $240 \div 8 - 1 = 29$ ）（1行の文字数30文字）

P5 = 48（表示の1フレーム時間の設定）

P6 = 64（1画面あたりの表示ライン数の設定）

P7 = 30（仮想スクリーンのアドレス数）

P8 = 00（ " ）

SCROLL

C = 01000100B

P1 = 00H (第1ブロック・メモリ・エリア0000H - 0FFFH)

P2 = 00H (開始アドレス)

P3 = 40H (第1ブロックの表示ライン数64本)

P4 = 00H (第2ブロック・メモリ・エリア1000H - 1FFFH)

P5 = 10H (開始アドレス)

P6 = 40H (第2ブロックの表示ライン数64本)

HDOT SCROLL (水平方向ドット指定)

C = 01011010B

P1 = 0

OVLAY

C = 01011011B

P1 = 00000010B (第1ブロック ExOR 第2ブロック)

DISP ON/OFF

C = 01011000B

P1 = 01010110B (各ブロック表示オン, カーソル・ブリンク)

CSRW

C = 01001011B

P1 = 00H (カーソル・アドレス0000Hの設定)

P2 = 00H

CSRDIR

C = 01001100B (カーソル・シフト方向を右に設定)

MWRITE

C = 01000010B (データ書き込みの設定)

P1 = XXXXXXXX (書き込みデータの設定)

P2 = XXXXXXXX (書き込みデータの設定)

⋮

Pn = XXXXXXXX (書き込みデータの設定)

第3章 機能説明

第3章では、電子手帳の機能を次に示すモードごとに、各キー入力による動作やモード遷移について説明します。

- ・キー入力
- ・メニュー
- ・名刺（個人）
- ・名刺（会社）
- ・カレンダー
- ・メモ帳
- ・世界時間
- ・初期設定
- ・通信

3.1 キー入力

電子手帳で使用するキーは、英字、数字、カーソル、記号、制御キーなどの合計59個があります。各キーの主な用途を表3 - 1に示します。

表3 - 1 電子手帳の使用キー一覧

キー名称	用途
A - Z (英字)	データ入力
0 - 9 (数字)	データ入力
(スペース), (,)	データ入力
, , ,	モード選択, カーソル移動
=, +, -, ×, ÷, .	データ入力
ENT	モード遷移実行, データ登録
DEL	データ削除, 文字削除
CLR	実行取消
SHIFT	キー・コード変更
UP, DOWN	参照データのページ変更
MENU	メニュー・モードに遷移
COMM	データ通信モードの選択
ON, OFF	パワーON/OFF制御

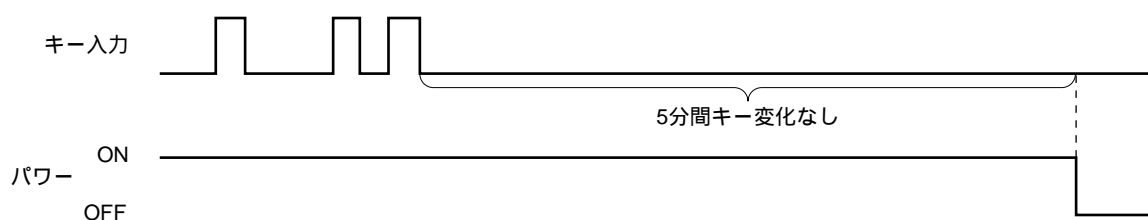
(1) キー入力方法

キー入力は、SHIFT、ONキーを除き、連続押し、多重押しを無効にするロック・アウト方式を用いています。ONキーは、OFF状態のときに有効になり、ON以外のキーは、ON状態のときに有効になります。SHIFTキーは、A - Zキーと同時に押すと、小文字の英字 (a - z) 入力になります。

(2) オートパワーOFF

電子手帳がONのときに、5分間キー入力に変化しなかったときは、自動的にパワーOFF状態になります。

図3 - 1 オートパワーOFF



3.2 メニュー・モード

パワーON,またはMENUキーの入力や,各モードからの復帰により,メニュー・モードになります。メニュー・モードでは,起動するモードの選択を行います。モードの選択方法は, , , ENTキーを使用して選択するか(反転表示が対象モード),メニュー画面の左端の数字(1-6)のキーを入力して選択します。メニュー・モードの表示画面を図3-2に,有効キーを表3-2に示します。

メニュー・モードでは,COMMキーを入力することにより,名刺(個人,会社),メモ帳データの全データを通信することができます。

図3-2 メニュー表示

	1	5	10	15	20	25	30										
1	M	E	N	U													
2	1	A	D	D	R	E	S	S	(M	A	N)				
3	2	A	D	D	R	E	S	S	(C	O	M	P	A	N	Y)
4	3	C	A	L	E	N	D	A	R								
5	4	M	E	M	O												
6	5	C	L	O	C	K											
7	6	S	E	T													
8																	

表3-2 メニュー・モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
1-6	対象のモードに移行
	反転表示行を1行上にする(2行目のときは変化なし)
	反転表示行を1行下にする(7行目のときは変化なし)
ENT	反転表示行のモードに移行
COMM	通信モードに移行

3.3 名刺（個人）機能

メニュー・モードで“1 ADDRESS (MAN)”を選択すると名刺（個人）モードになります。

名刺（個人）モードでは、最初に起動するモードの選択を行います。モードの選択は、、キーにより行い（反転表示が対象モード）、ENTキーにより決定します。名刺（個人）モードには、名刺データの登録、修正、検索、削除の4つのモードがあります。モード選択の表示画面を図3 - 3に、有効キーを表3 - 3に示します。

名刺（個人）モードでは、COMMキーを入力することにより、名刺（個人）の全データを通信することができます。

図3 - 3 名刺（個人）モード選択表示

	1	5	10	15	20	25	30					
1	A	D	D	R	E	S	S	(M	A	N)
2	REGISTER											
3	C	O	R	R	E	C	T	I	O	N		
4	S	E	A	R	C	H						
5	D	E	L	E	T	E						
6												
7												
8												

表3 - 3 名刺（個人）モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（5行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行
COMM	通信モードに移行

3.3.1 登録モード

登録モードは、モード選択のときにREGISTERを反転表示に設定し、ENTキーを入力します。登録モードを選択すると、最初に名前を入力するモードになり、ENTキーの入力でデータが決定されます。ENTキーの入力ごとに電話、FAX、住所、会社名、メモ内容の入力モードになります。メモ内容入力後のENTキーによりデータが登録されます。入力項目は、キーが入力されるまでLCDに表示され、データを入力すると消えます。登録モードの表示画面を図3-4に、有効キーを表3-4に示します。

図3-4 名刺(個人)登録モードの表示

	1	5	10	15	20	25	30
1	N	A	M	E	?		
2	T	E	L	?			
3	F	A	X	?			
4	A	D	D	R	E	S	S
5							
6	C	O	M	P	A	N	Y
7	M	E	M	O	?		
8							

表3-4 名刺(個人)登録モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする(5, 8行で有効)
	カーソルを下を表示行にする(4, 7行で有効)
	カーソルを右の表示桁にする(30桁目のときは次の行の1桁目にする)
	カーソルを左の表示桁にする(1桁目のときは前の行の30桁目にする)
文字データ ^注	データ入力
DEL	文字データ削除
SHIFT + DEL	文字データ1行削除
CLR	前画面に移行
ENT	<ul style="list-style-type: none"> 各入力内容の決定 MEMO?後のENT入力で、データを登録し前画面に移行

注 A-Z, 0-9, (,), +, -, *, /, =, (スペース), SHIFT + A-Z = a-z

3.3.2 データ選択モード

モード選択のときに、CORRECTION/SEARCH/DELETEを選択すると、データ選択モードになります。データ選択モードでは、修正/検索/削除したい名刺(個人)データの選択を行います。データ選択後は、データ表示をし、修正/検索/削除モードになります。

データの選択方法には、名前の先頭文字(カーソル表示)による検索方法と、先頭ページから順番に検索する方法の2種類があります。データの選択方法について次に示します。

(1) 先頭文字選択モード

データ選択モードは、まず先頭文字選択モード画面になり、A-ZとNEXT PAGEが表示され、Aの位置にカーソルがある状態で始まります。

名前の先頭文字から検索する場合は、**←**、**→**、**↑**、**↓**キーによりカーソルを移動して、検索したい名刺(個人)データの名前の先頭文字(A-Z)を選択し、ENTキー入力で決定します。先頭文字を決定すると、検索データ選択モードになります。

先頭ページから検索する場合は、**↓**キーでNEXT PAGEを反転表示に設定します。ENTキー入力によりデータ表示を行います。

先頭文字選択モードの表示画面を図3-5に、有効キーを表3-5に示します。

図3-5 名刺(個人)先頭文字選択モードの表示

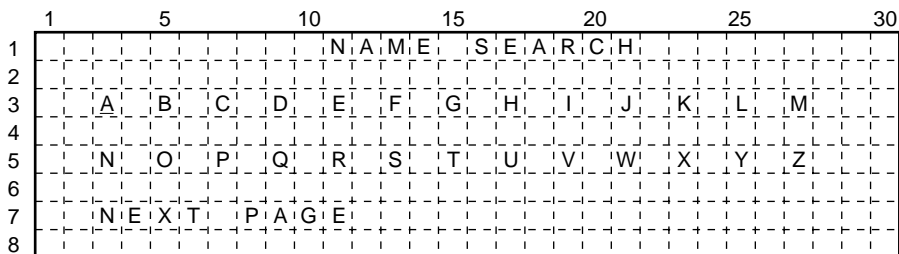


表3-5 名刺(個人)先頭文字選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする(A-Mでは変化なし)
	カーソルを下を表示行にする(NEXT PAGEでは変化なし)
	カーソルを右の表示桁にする(MのときはNへ、ZのときはNEXT PAGEへ移行する。NEXT PAGEでは変化なし)
	カーソルを左の表示桁にする(NのときはMへ、NEXT PAGEのときはZへ移行する。Aでは変化なし)
CLR	前画面に移行
ENT	<ul style="list-style-type: none"> 検索文字データ(A-Z)を決定し、検索データ選択モードに移行 NEXT PAGEのときはデータを表示

3.3.4 検索モード

検索モードは、モード選択のときにSEARCHを反転表示に設定し、ENTキーを入力します。

検索モードを選択すると、まずデータ選択モードになるので、検索したい名刺（個人）データを選択します（3.3.2 データ選択モード参照）。検索するデータを決定すると、名刺（個人）データの表示を行います。また、データ表示中に、ENTキーを入力すると、会社データの表示をします。ただし、名刺（会社）データに一致する会社名が登録されていないときは、会社名のみを表示します。検索モードの表示画面を図3-8に、有効キーを表3-8に示します。また、会社データ表示中の表示画面を図3-9に、有効キーを表3-9に示します。

検索モードでは、COMMキーを入力することにより、名刺（個人）の表示中のデータを通信することができます。

図3-8 名刺（個人）検索モードの表示

		1	5	10	15	20	25	30
名前	→	1	X X X X X X X X	X				
電話番号	→	2	X X X X - X X	X X X X				
FAX	→	3	X X X X - X X	X X X X				
住所	→	4	X X X X X X X X	X X X - X X - X X				
		5	X X X X X X X X	X				
会社メモ	→	6	X X X X X X X X					
		7	X X X X X X X X					
		8	X X X X X X X X					

表3-8 名刺（個人）検索モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
UP	次ページの名刺（個人）を表示（最終ページのときは変化なし）
DOWN	前ページの名刺（個人）を表示（先頭ページのときは変化なし）
CLR	前画面に移行
ENT	会社のデータを表示
COMM	通信モードに移行

図3-9 名刺(個人)会社データ一覧の表示

		1	5	10	15	20	25	30
会社	→	1	X X X X X X X X					
電話番号	→	2	X X X X X X X X					
FAX	→	3	X X X X X X X X					
住所	→	4	X X X X X X X X	X X X X X X X X				
		5	X X X X X X X X					
メモ	→	6	X X X X X X X X					
		7	X X X X X X X X					
		8	X X X X X X X X					

表3-9 名刺(個人)会社データ一覧の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行

3.3.5 削除モード

削除モードは、モード選択のときにDELETEを反転表示に設定し、ENTキーを入力します。削除モードを選択すると、まずデータ選択モードになるので、削除したい名刺(個人)データを選択します(3.3.2 データ選択モード参照)。先頭文字による削除データ検索中にDELキーを入力すると、反転表示している名前の名刺(個人)データが削除されます。また、データ表示中に、DELキーを入力すると、表示中の名刺(個人)データがすべて削除されます。削除モードの表示画面を図3-10に、有効キーを表3-10に示します。

図3-10 名刺(個人)削除モードの表示

		1	5	10	15	20	25	30
名前	→	1	X X X X X X X X					
電話番号	→	2	X X X X X X X X					
FAX	→	3	X X X X X X X X					
住所	→	4	X X X X X X X X	X X X X X X X X				
		5	X X X X X X X X					
会社	→	6	X X X X X X X X					
メモ	→	7	X X X X X X X X					
		8	X X X X X X X X					

表3-10 名刺(個人)削除モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行
UP	次ページの名刺(個人)データを表示(最終ページのときは変化なし)
DOWN	前ページの名刺(個人)データを表示(先頭ページのときは変化なし)
DEL	表示データを削除し、前画面に移行

3.4 名刺（会社）機能

メニュー・モードで“2 ADDRESS (COMPANY)”を選択すると名刺（会社）モードになります。

名刺（会社）モードでは、最初に起動するモードの選択を行います。モードの選択は、、 キーにより行い（反転表示が対象モード）、ENTキーにより決定します。名刺（会社）モードには、名刺データの登録、修正、検索、削除の4つのモードがあります。モード選択の表示画面を図3 - 11に、有効キーを表3 - 11に示します。

名刺（会社）モードでは、COMMキーを入力することにより、名刺（会社）の全データを通信することができます。

図3 - 11 名刺（会社）モード選択表示

	1	5	10	15	20	25	30									
1	A	D	D	R	E	S	S	(C	O	M	P	A	N	Y)
2	REGISTER															
3	C	O	R	R	E	C	T	I	O	N						
4	S	E	A	R	C	H										
5	D	E	L	E	T	E										
6																
7																
8																

表3 - 11 名刺（会社）モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（5行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行
COMM	通信モードに移行

3.4.1 登録モード

登録モードは、モード選択のときにREGISTERを反転表示に設定し、ENTキーを入力します。登録モードを選択すると、最初に会社名を入力するモードになり、ENTキーの入力でデータが決定されます。ENTキー入力ごとに電話、FAX、住所、メモ内容の入力モードになります。メモ内容入力後のENTキーによりデータが登録されます。入力項目は、キーが入力されるまでLCDに表示されます。登録モードの表示画面を図3-12に、有効キーを表3-12に示します。

図3-12 名刺(会社)登録モードの表示

	1	5	10	15	20	25	30	
1	C	O	M	P	A	N	Y	?
2	T	E	L	?	-	-	-	-
3	F	A	X	?	-	-	-	-
4	A	D	D	R	E	S	S	?
5	-	-	-	-	-	-	-	-
6	M	E	M	O	?	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-

表3-12 名刺(会社)登録モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする(5, 7, 8行で有効)
	カーソルを下を表示行にする(4, 6, 7行で有効)
	カーソルを右の表示桁にする(30桁目のときは次の行の1桁目にする)
	カーソルを左の表示桁にする(1桁目のときは前の行の30桁目にする)
文字データ ^注	データ入力
DEL	文字データ削除
SHIFT + DEL	文字データ1行削除
CLR	前画面に移行
ENT	<ul style="list-style-type: none"> 各入力内容の決定 MEMO?後のENT入力で、データを登録し前画面に移行

注 A-Z, 0-9, (,), +, -, *, /, =, (スペース), SHIFT + A-Z = a-z

3.4.2 データ選択モード

モード選択のときに、CORRECTION/SEARCH/DELETEを選択すると、データ選択モードになります。データ選択モードでは、修正/検索/削除したい名刺（会社）データの選択を行います。データ選択後は、データ表示をし、修正/検索/削除モードになります。

データの選択方法には、会社名の先頭文字（カーソル表示）による検索方法と、先頭ページから順番に検索する方法の2種類があります。データの選択方法について次に示します。

(1) 先頭文字選択モード

データ選択モードは、まず先頭文字選択モード画面になり、A-ZとNEXT PAGEが表示され、Aの位置にカーソルがある状態で始まります。

会社名の先頭文字から検索する場合は、
、
、
キーによりカーソルを移動して、検索したい名刺（会社）データの会社名の先頭文字（A-Z）を選択し、ENTキー入力で決定します。先頭文字を決定すると検索データ選択モードになります。

先頭ページから検索する場合は、
キーでNEXT PAGEを反転表示に設定します。ENTキー入力によりデータ表示を行います。

先頭文字選択モードの表示画面を図3-13に、有効キーを表3-13に示します。

図3-13 名刺（会社）先頭文字選択モードの表示

	1	5	10	15	20	25	30										
1				N	A	M	E	S	E	A	R	C	H				
2																	
3		A	B	C	D	E	F	G	H	I	J	K	L	M			
4																	
5		N	O	P	Q	R	S	T	U	V	W	X	Y	Z			
6																	
7		N	E	X	T	P	A	G	E								
8																	

表3-13 名刺（会社）先頭文字選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする（A-Mでは変化なし）
	カーソルを下を表示行にする（NEXT PAGEでは変化なし）
	カーソルを右の表示桁にする（MのときはNへ、ZのときはNEXT PAGEへ移行する。NEXT PAGEでは変化なし）
	カーソルを左の表示桁にする（NのときはMへ、NEXT PAGEのときはZへ移行する。Aでは変化なし）
CLR	前画面に移行
ENT	・検索文字データ（A-Z）を決定し、検索データ選択モードに移行 ・NEXT PAGEのときはデータを表示

(2) 検索データ選択モード

検索データ選択モードでは、選択された文字（サーチ・データ）で始まる会社名の一覧を表示します。次に **↑**、**↓**、**←**、**→** キーにより検索データを選択し（反転表示が対象データ）、ENTキー入力により反転表示している会社名の名刺（会社）データを表示します。また、UP/DOWNキーを入力することにより、先頭文字が変更されます。検索データ選択モードの表示画面を図3-14に、有効キーを表3-14に示します。

図3-14 名刺（会社）検索データ選択モードの表示

	1	5	10	15	20	25	30
1	N A M E D A T A (A)						
2	A x x x x						
3	A x x x x						
4	A x x x x						
5	A x x x x						
6	A x x x x						
7	A x x x x						
8	A x x x x						

備考 先頭文字がサーチ・データ（この場合はA）の会社名を表示（x：文字データ）

表3-14 名刺（会社）検索データ選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（最下位行のときは変化なし）
	会社名一覧表示を次ページにする（最終ページのときは変化なし）
	会社名一覧表示を前ページにする（先頭ページのときは変化なし）
CLR	前画面に移行
ENT	検索する名刺（会社）データを決定し、データを表示
UP	サーチ・データを次の文字に変更（Aのときは変化なし）
DOWN	サーチ・データを前の文字に変更（Zのときは変化なし）
DEL	反転表示行のデータを削除（削除モードのみ有効）

3.4.3 修正モード

修正モードは、モード選択のときにCORRECTIONを反転表示に設定し、ENTキーを入力します。修正モードを選択すると、まずデータ選択モードになるので、修正したい名刺（会社）データを選択します（3.4.2 データ選択モード参照）。修正するデータを決定すると、その名刺（会社）データ表示画面になりカーソルが2行目1桁目に表示されるので、**←**、**→**、**↑**、**↓** キーによりカーソルを移動します。次に文字データを入力すると、カーソルの位置から上書きされます。修正終了後は、ENTキー入力によりデータが再登録されます。修正モードの表示画面を図3 - 15に、有効キーを表3 - 15に示します。

図3 - 15 名刺（会社）修正モードの表示

		1	5	10	15	20	25	30
会社	→	1	X X X X X X X X					
電話番号	→	2	X X X X - X X - X X X X					
FAX	→	3	X X X X - X X - X X X X					
住所	→	4	X X X X X X X X X X X X					
		5	X X X X X X X X X X X X					
メモ	→	6	X X X X X X X X X X X X					
		7	X X X X X X X X X X X X					
		8	X X X X X X X X X X X X					

表3 - 15 名刺（会社）修正モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上の表示行にする（1行目のときは変化なし）
	カーソルを下の表示行にする（8行目のときは変化なし）
	カーソルを右の表示桁にする（30桁目のときは次の行の1桁目にする）
	カーソルを左の表示桁にする（1桁目のときは前の行の30桁目にする）
文字データ ^注	入力文字のデータをカーソルの場所に上書きする
DEL	文字データ削除
SHIFT + DEL	文字データ1行削除
UP	次ページの名刺（会社）を表示（最終ページのときは変化なし）
DOWN	前ページの名刺（会社）を表示（先頭ページのときは変化なし）
CLR	前画面に移行（修正データを無効にする）
ENT	修正データを登録し、前画面に移行

注 A-Z, 0-9, (,), +, -, *, /, =, (スペース), SHIFT + A-Z = a-z

3.4.4 検索モード

検索モードは、モード選択のときにSEARCHを反転表示に設定し、ENTキーを入力します。

検索モードを選択すると、まずデータ選択モードになるので、検索したい名刺（会社）データを選択します（3.4.2 データ選択モード参照）。検索するデータを決定すると、名刺（会社）データの表示を行います。また、データ表示中に、ENTキーを入力すると、表示中の会社名と、名刺（個人）データに登録されている会社名が、一致しているデータの名前を表示します（会社員一覧表示）。ただし、名刺（個人）データに一致する会社名が登録されていないときは、会社名のみを表示します。検索モードの表示画面を図3-16に、有効キーを表3-16に示します。また、会社員一覧の表示画面を図3-17に、有効キーを表3-17に示します。

検索モードでは、COMMキーを入力することにより、名刺（会社）の表示中のデータを通信することができます。

図3-16 名刺（会社）検索モードの表示

		1	5	10	15	20	25	30
会社	→	X	X	X	X	X	X	X
電話番号	→	X	X	X	X	X	X	X
FAX	→	X	X	X	X	X	X	X
住所	→	X	X	X	X	X	X	X
		X	X	X	X	X	X	X
メモ	→	X	X	X	X	X	X	X
		X	X	X	X	X	X	X
		X	X	X	X	X	X	X

表3-16 名刺（会社）検索モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
UP	次ページの名刺（会社）を表示（最終ページのときは変化なし）
DOWN	前ページの名刺（会社）を表示（先頭ページのときは変化なし）
CLR	前画面に移行
ENT	会社員一覧を表示
COMM	通信モードに移行

図3-17 名刺(会社)会社員一覧の表示

		1	5	10	15	20	25	30
会社	→	1	X X X X X X X X					
会社員	→	2		X X X X X X X				
		3	X X X X X X X					
		4		X X X X X X X				
		5	X X X X X X X					
		6	X X X X X X X					
		7	X X X X X X X					
		8	X X X X X X X					

表3-17 名刺(会社)会社員一覧の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行

3.4.5 削除モード

削除モードは、モード選択のときにDELETEを反転表示に設定し、ENTキーを入力します。削除モードを選択すると、まずデータ選択モードになるので、削除したい名刺(会社)データを選択します(3.4.2 データ選択モード参照)。先頭文字による削除データ検索中にDELキーを入力すると、反転表示している会社名の名刺(会社)データが削除されます。また、データ表示中に、DELキーを入力すると、表示中の名刺(会社)データがすべて削除されます。削除モードの表示画面を図3-18に、有効キーを表3-18に示します。

図3-18 名刺(会社)削除モードの表示

		1	5	10	15	20	25	30
会社	→	1	X X X X X X X X					
電話番号	→	2	X X X X X X X X					
FAX	→	3	X X X X X X X X					
住所	→	4	X X X X X X X X					
		5	X X X X X X X X					
メモ	→	6	X X X X X X X X					
		7	X X X X X X X X					
		8	X X X X X X X X					

表3-18 名刺(会社)削除モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行
UP	次ページの名刺(会社)データを表示(最終ページのときは変化なし)
DOWN	前ページの名刺(会社)データを表示(先頭ページのときは変化なし)
DEL	表示データを削除し、前画面に移行

3.5 カレンダー機能

メニュー・モードで“3 CALENDAR”を選択するとカレンダー・モードになります。

カレンダー・モードでは、最初に起動するモードの選択を行います。モードの選択は、**↑**、**↓**キーにより行い（反転表示が対象モード）、**ENT**キーにより決定します。カレンダー・モードには、現時刻のカレンダー表示、カレンダー表示の月設定、休日の設定の3つのモードがあります。モード選択の表示画面を図3-19に、有効キーを表3-19に示します。

図3-19 カレンダー・モード選択表示

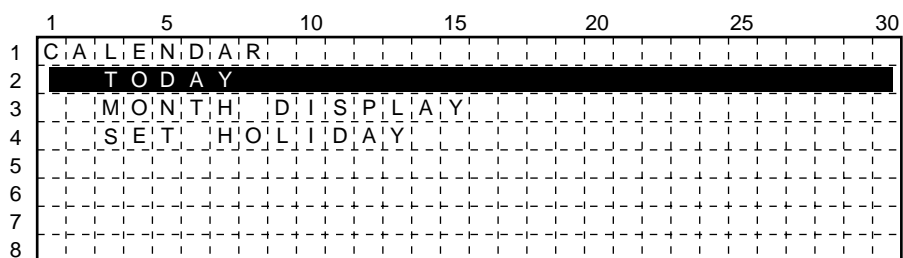


表3-19 カレンダー・モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（4行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行

3.5.1 現時刻のカレンダー表示

現時刻のカレンダー表示は、モード選択のときにTODAYを反転表示に設定し、ENTキーを入力します。

現時刻のカレンダー表示では、現時刻のカレンダーの月表示と、スケジュールの登録、表示（修正）、削除をします。現時刻のカレンダーの表示画面を図3-20に、有効キーを表3-20に示します。

図3-20 現時刻のカレンダー表示

	1	5	10	15	20	25	30
1				x	x	x	x
2	S	U	N	M	O	N	T
3							
4		2	3	4	(*	5
5	*	9	10	11	12	13	14
6		16	17	18	19	20	*
7	*	23	24	25	26	27	28
8		30	31				

表3-20 現時刻のカレンダー表示の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	(xx)を1行上にする(日付がないときは前月表示)
	(xx)を1行下にする(日付がないときは次月表示)
	(xx)を+1日する(1日のときは前月最終日)
	(xx)を-1日する(最終日のときは次月の1日)
UP	次の月を表示(2099年12月以降変化なし)
DOWN	前の月を表示(1901年1月以前変化なし)
SHIFT + UP	次の年を表示(2099年以降変化なし)
SHIFT + DOWN	前の年を表示(1901年以前変化なし)
CLR	前画面に移行
ENT	(xx)の日付のスケジュール・モード

3.5.2 カレンダー表示の月設定

カレンダー表示の月設定は、モード選択のときにMONTH DISPLAYを反転表示に設定し、ENTキーを入力します。

カレンダー表示の月設定では、カレンダーの表示年月を指定したあと、その月のカレンダー表示を行います。

(1) 表示月設定

カレンダー表示の月設定モードを選択すると、まず年入力モードになり、数値（1901 - 2099）入力後のENTキー入力で決定されます。ENTキー入力後は、月入力モードになり、数値（1 - 12）入力後のENTキーで決定され、カレンダー表示モードになります（図3 - 25参照）。カレンダー表示の月設定モードの表示画面を図3 - 24に、有効キーを表3 - 24に示します。

図3 - 24 カレンダー表示の月設定モードの表示

	1	5	10	15	20	25	30									
1																
2	Y	E	A	R	?	(1	9	0	1	-	2	0	9	9)
3	x	x	x	x												
4	M	O	N	T	H	?										
5	x	x														
6																
7																
8																

表3 - 24 カレンダー表示の月設定モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
0-9	データ入力
DEL	文字データ削除
CLR	前画面に移行
ENT	<ul style="list-style-type: none"> ・ ENT入力で、データを決定する ・ MONTHデータ入力後のENT入力でカレンダー表示をする

(2) 月設定時のカレンダー表示

月設定時のカレンダー表示は、表示月設定モードで入力された月のカレンダー表示をします。表示月の変更は、UP, DOWN, SHIFT + UP/DOWNキー入力により行います。月設定時のカレンダー表示の表示画面を図3 - 25に、有効キーを表3 - 25に示します。

図3 - 25 月設定時のカレンダー表示

	1	5	10	15	20	25	30
1				x	x	x	x
2	S	U	N	M	O	N	T
3		x	x		x	x	
4		x	x		x	x	
5		x	x		x	x	
6		x	x		x	x	
7		x	x		x	x	
8		x	x		x	x	

表3 - 25 月設定時のカレンダー表示の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
UP	次の月を表示 (2099年12月以降変化なし)
DOWN	前の月を表示 (1901年1月以前変化なし)
SHIFT + UP	次の年を表示 (2099年以降変化なし)
SHIFT + DOWN	前の年を表示 (1901年以前変化なし)
CLR	前画面に移行

3.5.3 休日の設定

休日の設定は、モード選択のときにSET HOLIDAYを反転表示に設定し、ENTキーを入力します。休日の設定には、毎週休日になる曜日を指定する週末モードと、1日単位で休日の指定をする日付指定モードの2種類があります。これらのモードの選択は、**←**、**→** キーにより行い（反転表示が対象モード）、ENTキーにより決定します。休日の設定モード選択の表示画面を図3 - 26、有効キーを表3 - 26に示します。

図3 - 26 休日の設定モード選択表示

	1	5	10	15	20	25	30			
1	S	E	T	H	O	L	I	D	A	Y
2	W E E K L Y H O L I D A Y									
3	S	E	T	D	A	T	E			
4	-----									
5	-----									
6	-----									
7	-----									
8	-----									

表3 - 26 休日の設定モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（3行目のときは変化なし）
CLR	前画面に移行
ENT	ENT入力で反転表示しているモードに移行

(1) 週休モード

週休モードは、モード選択のときにWEEKLY HOLIDAYを反転表示に設定し、ENTキーを入力します。、キーにより曜日を指定し、、キーにより休日/平日の設定をします。設定後のENTキー入力により、データが有効になります。週休モードの表示画面を図3-27に、有効キーを表3-27に示します。

図3-27 週休モードの表示

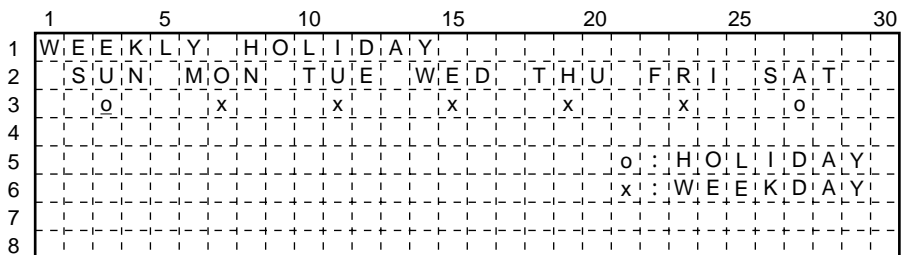


表3-27 週休モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルの曜日を休日に設定
	カーソルの曜日を平日に設定
	カーソルを右の表示桁にする (SATからはSUNへ移動)
	カーソルを左の表示桁にする (SUNからはSATへ移動)
CLR	前画面に移行 (表示内容は無効)
ENT	ENT入力で、データを登録し前画面に戻る

(2) 日付指定モード

日付指定モードは、モード選択のときにSET DATEを反転表示に設定し、ENTキーを入力します。

日付指定モードでは、休日にしたい日付を年、月、日の順に入力します。また、休日を解除する場合は、データ入力後DELキーを入力します。

日付指定モードを選択すると、まず年入力モードになり、数値(1901-2099)入力後のENTキー入力で決定されます。ENTキー入力後は、月、日の順で入力モードになり、数値(月:1-12, 日:1-31)入力後のENTキーで決定されます。日データ入力後のENTキーにより休日の指定になり、DELキーにより平日の指定になります。日付指定モードの表示画面を図3-28に、有効キーを表3-28に示します。

図3-28 日付指定モードの表示

	1	5	10	15	20	25	30		
1	S	E	T	D	A	T	E		
2	Y	E	A	R	?	x	x	x	x
3	M	O	N	T	H	?	x	x	
4	D	A	Y	?	x	x			
5									
6									
7									
8									

表3-28 日付指定モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを右の表示桁にする
	カーソルを左の表示桁にする
0-9	データ入力
DEL	・文字データ削除 ・DAYデータ入力後のDEL入力で、指定した日付を平日にし(休日解除),前画面に戻る
CLR	前画面に移行
ENT	・各入力内容の決定 ・DAYデータ入力後のENT入力で、指定した日付を休日として登録し前画面に戻る

3.6 メモ帳機能

メニュー・モードで“4 MEMO”を選択するとメモ帳モードになります。

メモ帳モードでは、最初に起動するモードの選択を行います。モードの選択は、、キーにより行い（反転表示が対象モード）、ENTキーにより決定します。メモ帳モードには、メモ帳データの登録、修正、検索、削除の4つのモードがあります。モード選択の表示画面を図3 - 29に、有効キーを表3 - 29に示します。

メモ帳モードでは、COMMキーを入力することにより、メモ帳の全データを通信することができます。

図3 - 29 メモ帳モード選択表示

	1	5	10	15	20	25	30			
1	M	E	M	O						
2	REG I S T E R									
3	C	O	R	R	E	C	T	I	O	N
4	S	E	A	R	C	H				
5	D	E	L	E	T	E				
6										
7										
8										

表3 - 29 メモ帳モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（5行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行
COMM	通信モードに移行

3.6.1 登録モード

登録モードは、モード選択のときにREGISTERを反転表示に設定し、ENTキーを入力します。登録モードでは、まずレベル1のタイトル（10文字以内。11文字以降のデータは無効）を作成します。次にレベル2のタイトル（30文字以内）を作成し（REGISTER（LEVEL2））、そのタイトルのページ内にデータを入力します。また、関連のあるメモ内容をレベル1の同一タイトルでまとめることで、ディレクトリ管理をします。

登録モードを選択すると、最初にレベル1のタイトルを入力するモードになり、ENTキーの入力でデータが決定されます。ENTキー入力後は、レベル2のタイトル入力モードになり、ENTキーの入力でデータが決定され、メモ帳の入力モードになります。メモ内容入力後のENTキーによりデータが登録されます。登録モード（タイトル、データ）の表示画面を図3-30、3-31に、有効キーを表3-30、3-31に示します。

図3-30 メモ帳タイトル登録モードの表示

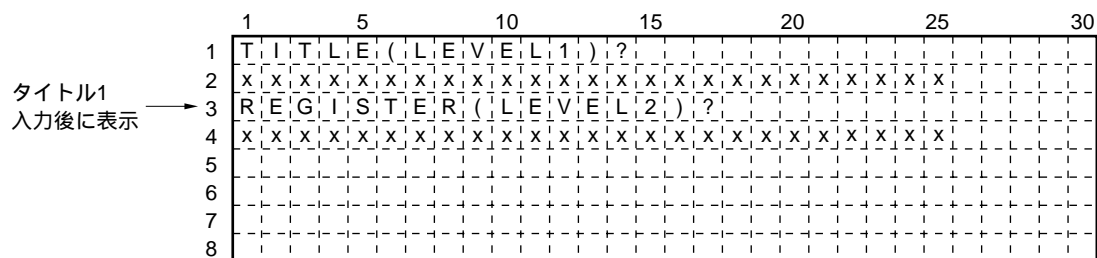


表3-30 メモ帳タイトル登録モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを右の表示桁にする
	カーソルを左の表示桁にする
文字データ ^注	データ入力
DEL	文字データ削除
SHIFT + DEL	文字データ1行削除
CLR	前画面に移行
ENT	タイトルを登録する（レベル2登録モード）

注 A-Z, 0-9, (,), +, -, *, /, =, (スペース), SHIFT + A-Z = a-z

図3-31 メモ帳データ登録モードの表示

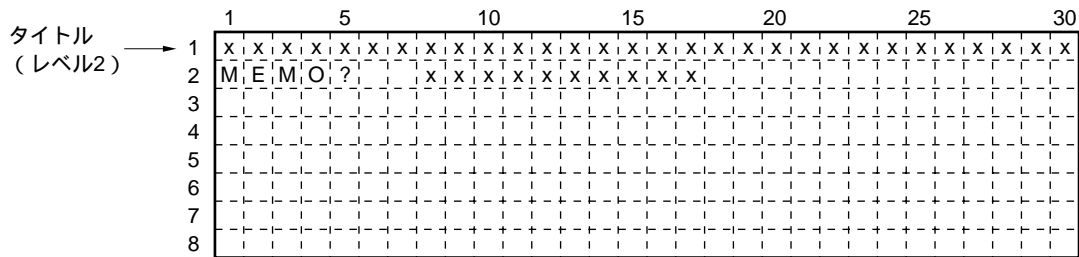


表3-31 メモ帳データ登録モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする (1行目のときは変化なし)
	カーソルを下を表示行にする (8行目のときは変化なし)
	カーソルを右の表示桁にする (30桁目のときは次の行の1桁目にする)
	カーソルを左の表示桁にする (1桁目のときは前の行の30桁目にする)
文字データ ^注	データ入力
DEL	文字データ削除
SHIFT + DEL	文字データ1行削除
CLR	前画面に移行 (データ無効)
ENT	MEMO?後のENT入力で、データを登録し前画面に移行

注 A-Z, 0-9, (,), +, -, *, /, =, (スペース), SHIFT + A-Z = a-z

3.6.2 データ選択モード

モード選択のときに、CORRECTION/SEARCH/DELETEを選択すると、データ選択モードになります。データ選択モードでは、修正/検索/削除したいメモ帳データの選択を行います。データ選択後は、データ表示をし、修正/検索/削除モードになります。

データの選択方法には、タイトルのレベル1の先頭文字（カーソル表示）によるデータ名の検索方法と、先頭ページから順番に検索する方法の2種類があります。データの選択方法について次に示します。

(1) 先頭文字選択モード

データ選択モードは、まず先頭文字選択モード画面になりA-ZとNEXT PAGEが表示され、Aの位置にカーソルがある状態で始まります。

タイトル（レベル1）の先頭文字から検索する場合は、
、
、
キーによりカーソルを移動して、検索したいメモ帳データのタイトル（レベル1）の先頭文字（A-Z）を選択し、ENTキー入力で決定します。先頭文字を決定するとタイトル（レベル1）選択モードになります。

先頭ページから検索する場合は、
キーでNEXT PAGEを反転表示に設定します。ENTキー入力によりデータ表示を行います。

先頭文字選択モードの表示画面を図3-32に、有効キーを表3-32に示します。

図3-32 メモ帳タイトル先頭文字選択モードの表示

	1	5	10	15	20	25	30
1	T I T L E S E A R C H						
2							
3	A	B	C	D	E	F	G
4	H	I	J	K	L	M	
5	N	O	P	Q	R	S	T
6	U	V	W	X	Y	Z	
7	N E X T P A G E						
8							

表3-32 メモ帳タイトル先頭文字選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを上を表示行にする（A-Mでは変化なし）
	カーソルを下を表示行にする（NEXT PAGEでは変化なし）
	カーソルを右の表示桁にする（MのときはNへ、ZのときはNEXT PAGEへ移行する。NEXT PAGEでは変化なし）
	カーソルを左の表示桁にする（NのときはMへ、NEXT PAGEのときはZへ移行する。Aでは変化なし）
CLR	前画面に移行
ENT	・ 検索文字データ（A-Z）を決定し、タイトル（レベル1）選択モードに移行 ・ NEXT PAGEのときはデータを表示

(3) タイトル(レベル2) 選択モード

タイトル(レベル2) 選択モードでは、選択されたタイトル(レベル1) に登録されているレベル2 のタイトル一覧を表示します。次に , , キーにより反転表示のデータを変更し、タイトル (レベル2) を選択します。ENTキー入力により反転表示しているタイトル(レベル2) のデータを表示します。タイトル(レベル2) 選択モードの表示画面を図3 - 34に、有効キーを表3 - 34に示します。

図3 - 34 タイトル(レベル2) 選択モードの表示

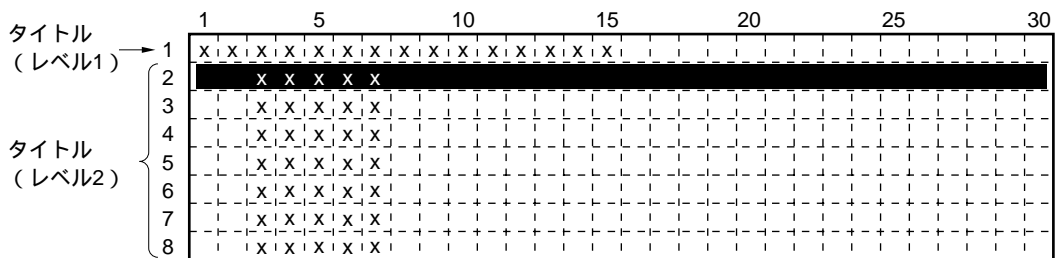


表3 - 34 タイトル(レベル2) 選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする(2行目のときは変化なし)
	反転表示行を1行下にする(最下位行のときは変化なし)
	タイトル(レベル2) の表示を次ページにする(最終ページのときは変化なし)
	タイトル(レベル2) の表示を前ページにする(先頭ページのときは変化なし)
CLR	前画面に移行
ENT	検索するタイトル(レベル2) を決定し、データを表示
DEL	反転表示行のデータをすべて削除し、前画面に移行(削除モードのみ有効)

3.6.5 削除モード

削除モードは、モード選択のときにDELETEを反転表示に設定し、ENTキーを入力します。削除モードを選択後、先頭文字による削除データ検索中にDELキーを入力すると、反転表示しているタイトル内のデータがすべて削除されます。また、データ表示中に、DELキーを入力すると、表示中のメモ帳データがすべて削除されます。削除モードの表示画面を図3-37に、有効キーを表3-37に示します。

図3-37 メモ帳削除モードの表示

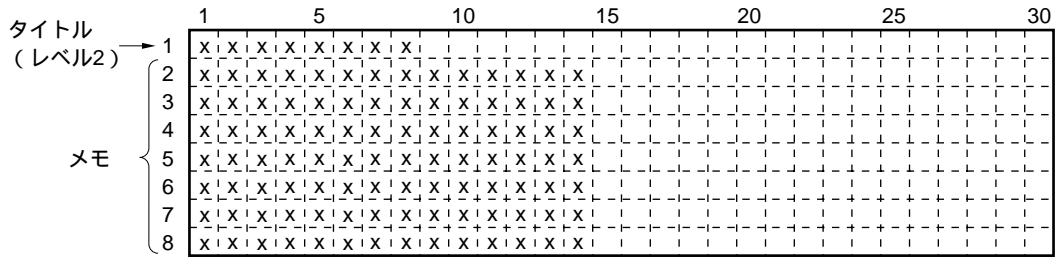


表3-37 メモ帳削除モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行
UP	次ページのメモ帳データを表示（最終ページのときは変化なし）
DOWN	前ページのメモ帳データを表示（先頭ページのときは変化なし）
DEL	表示データを削除し、前画面に移行

3.7 世界時計機能

メニュー・モードで“5 CLOCK”を選択すると世界時計モードになります。

世界時計モードでは、最初に起動するモードの選択を行います。モードの選択は、**←**、**→** キーにより行い（反転表示が対象モード）、**ENT**キーにより決定します。世界時計モードには、時計表示と、時計表示の都市選択の2つのモードがあります。世界時計モード選択の表示画面を図3-38に、有効キーを表3-38に示します。

図3-38 世界時計モード選択表示

	1	5	10	15	20	25	30			
1	C	L	O	C	K					
2	DISP L A Y									
3	C	I	T	Y	S	E	L	E	C	T
4	-----									
5	-----									
6	-----									
7	-----									
8	-----									

表3-38 世界時計モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（3行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行

3.7.1 時計表示モード

時計表示モードは、モード選択のときにDISPLAYを反転表示に設定し、ENTキーを入力します。時計表示モードを選択すると、都市名、日付、時間の表示をします。表示形式は表示画面を2分割し、左側に東京の時間を、右側に選択した都市（初期設定はニューヨーク。都市の選択方法は3.7.2 時計表示の都市選択モードを参照）の時間を表示します。時計表示モードの表示画面を図3-39に、有効キーを表3-39に示します。

図3-39 時計表示モードの表示

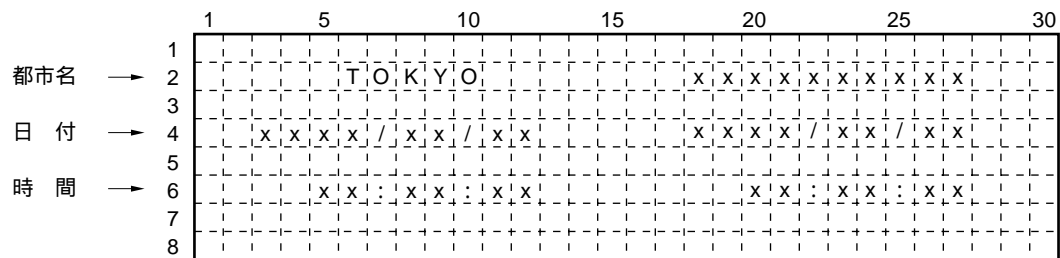


表3-39 時計表示モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
CLR	前画面に移行

3.7.2 時計表示の都市選択モード

時計表示の都市選択モードは、モード選択のときにCITY SELECTを反転表示に設定し、ENTキーを入力します。時計表示の都市選択モードでは、**←**、**→**、UP、DOWNキー入力により、メニュー表示された世界都市14箇所から1箇所を選択します（図3-40参照）。選択後は、モード選択画面になります。時計表示モードにすると、画面の右側に選択した都市の時計表示をします（図3-39参照）。時計表示の都市選択モードの表示画面を図3-40に、有効キーを表3-40に示します。

図3-40 時計表示の都市選択モードの表示

(a) 1ページ目の選択都市

	1	5	10	15	20	25	30							
1	C	I	T	Y	S	E	L	E	C	T				
2	NEW YORK													
3	L	O	S	A	N	G	E	L	E	S				
4	B	R	A	S	I	L	I	A						
5	L	I	M	A										
6	L	O	N	D	O	N								
7	R	O	M	E										
8	C	A	I	R	O									

(b) 2ページ目の選択都市

	1	5	10	15	20	25	30							
1	C	I	T	Y	S	E	L	E	C	T				
2	MOSCOW													
3	H	O	N	O	L	U	L	U						
4	S	Y	D	N	E	Y								
5	P	E	K	I	N	G								
6	H	O	N	G	K	O	N	G						
7	N	E	W	D	E	L	H	I						
8	K	U	W	A	I	T								

表3-40 時計表示の都市選択モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（8行目のときは変化なし）
UP	2ページ目の都市データを表示（1ページ目で有効）
DOWN	1ページ目の都市データを表示（2ページ目で有効）
CLR	前画面に移行（修正データを無効にする）
ENT	反転表示行を表示都市に選択し、前画面に移行

3.8 初期設定

メニュー・モードで“6 SET”を選択すると初期設定モードになります。

初期設定モードでは、最初に起動するモードの選択を行います。モードの選択は、**↑**、**↓** キーにより行い（反転表示が対象モード）、**ENT**キーにより決定します。初期設定モードには、日付設定と、時間設定の2つのモードがあります。モード選択の表示画面を図3 - 41に、有効キーを表3 - 41に示します。

図3 - 41 初期設定モード選択表示

	1	5	10	15	20	25	30
1	S	E	T				
2	DATE						
3	T	I	M	E			
4							
5							
6							
7							
8							

表3 - 41 初期設定モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2行目のときは変化なし）
	反転表示行を1行下にする（3行目のときは変化なし）
CLR	前画面に移行
ENT	反転表示行のモードに移行

3.8.1 日付設定モード

日付設定モードは、モードの選択のときにDATEを反転表示に設定し、ENTキーを入力します。日付設定モードを選択すると、まず年入力モードになり、数値（1901 - 2099）入力後のENTキー入力で決定されます。ENTキー入力後は、月、日の順で入力モードになり、数値（月：1 - 12、日：1 - 31）入力後のENTキーで決定されます。日データ入力後のENTキーにより、設定データが有効になります。データを何も入力しないでENTキーを入力した場合は、年設定データは2000に、月、日設定データは1になります。日付設定モードの表示画面を図3 - 42に、有効キーを表3 - 42に示します。

図3 - 42 日付設定モードの表示

	1	5	10	15	20	25	30
1	S	E	T	D	A	T	E
2	x	x	x	x	/	x	x
3	Y	E	A	R	?		
4	x	x	x	x			
5	M	O	N	T	H	?	
6	x	x					
7	D	A	Y	?			
8	x	x					

表3 - 42 日付設定モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを右の表示桁にする
	カーソルを左の表示桁にする
0 - 9	データ入力
DEL	文字データ削除
CLR	前画面に移行（入力データは無効）
ENT	<ul style="list-style-type: none"> 各入力内容の決定 DAY?後のENT入力で、データを登録し前画面に移行

3.8.2 時間設定モード

時間設定モードは、モード選択のときにTIMEを反転表示に設定し、ENTキーを入力します。日付設定モードを選択すると、まず時入力モードになり、数値(00-23)入力後のENTキー入力で決定されます。ENTキー入力後は、分、秒の順で入力モードになり、数値(00-59)入力後のENTキーで決定されます。秒データ入力後のENTキーにより、設定データが有効になります。データを何も入力しないでENTキーを入力した場合は、設定データは00になります。時間設定モードの表示画面を図3-43に、有効キーを表3-43に示します。

図3-43 時間設定モードの表示

	1	5	10	15	20	25	30
1	S	E	T	T	I	M	E
2		x	x		x	x	
3		H	O	U	R	?	
4			x	x			
5		M	I	N	U	T	E
6			x	x			
7		S	E	C	O	N	D
8			x	x			

表3-43 時間設定モードの有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	カーソルを右の表示桁にする
	カーソルを左の表示桁にする
0-9	データ入力
DEL	文字データ削除
CLR	前画面に移行(入力データは無効)
ENT	<ul style="list-style-type: none"> 各入力内容の決定 SECOND?後のENT入力で、データを登録し前画面に移行

3.9 通信機能

名刺（個人、会社）モードとメモ帳モードのモード選択，検索モードのときおよび，メニュー・モードのときにCOMMキーを入力すると通信モードになります。

通信モードでは，名刺（個人、会社）データやメモ帳データの送受信を，PC - 9801シリーズと，または電子手帳同士で行います。PC - 9801シリーズとの通信はRS - 232Cを介して行い，電子手帳同士の通信はμPD78014の3線式モードを使用して行います。通信モードでは，まず送受信モード（送信/受信）の設定を行い，次に通信先モード（PC - 9801モード/電子手帳モード）の設定を行います。通信モードの表示画面を図3 - 44に，有効キーを表3 - 44に示します。

送受信モードの設定

送受信モードの設定は，**↑**，**↓** キーにより行い（反転表示が対象モード），ENTキーにより決定します。送信モードのときはTRANSMITを反転表示に，受信モードのときはRECEIVEを反転表示にして，ENTキーを入力します。ENTキー入力後は通信先モードの設定になります。

通信先モードの設定

通信先モードの設定は，**↑**，**↓** キーにより行い（反転表示が対象モード），ENTキーにより決定します。PC - 9801モードのときはPC - 9801（RS - 232C）を反転表示に，電子手帳モードのときはELECTRONIC POCKETBOOKを反転表示にして，ENTキーを入力します。ENTキー入力後，通信を開始し，通信が終了するとCOMMキー入力前の表示画面に戻ります。また，通信中にエラーが発生したときは，通信を中止し，エラー表示（ERROR）をします。

図3 - 44 通信モード選択表示

	1	5	10	15	20	25	30
1	C O M M U N I C A T I O N						
2	T R A N S M I T						
3	R E C E I V E						
4	-						
5	-						
送受信モード 選択後に表示 →	P C - 9 8 0 1 (R S - 2 3 2 C)						
7	E L E C T R O N I C P O C K E T B O O K						
8	-						

表3 - 44 通信モード選択の有効キー

有効キー	動作内容
OFF	パワーOFF状態
MENU	メニュー・モードに移行
	反転表示行を1行上にする（2または6行目のときは変化なし）
	反転表示行を1行下にする（3または7行目のときは変化なし）
CLR	前画面に移行
ENT	通信開始

3.9.1 電子手帳とPC - 9801シリーズの通信方法

(1) 入力形式

電子手帳 PC - 9801シリーズの通信

A : ¥ > CPYA AUX filename

PC - 9801シリーズ 電子手帳の通信

A : ¥ > CPYA filename AUX

(2) 出力メッセージ

正常終了した場合

finished transmission!!

入力形式に誤りがあった場合

Illegal input type

ファイルが作れない, または開けない場合

Could not create filename

Could not find filename

通信エラーが発生した場合

Aborted transmission by error!!

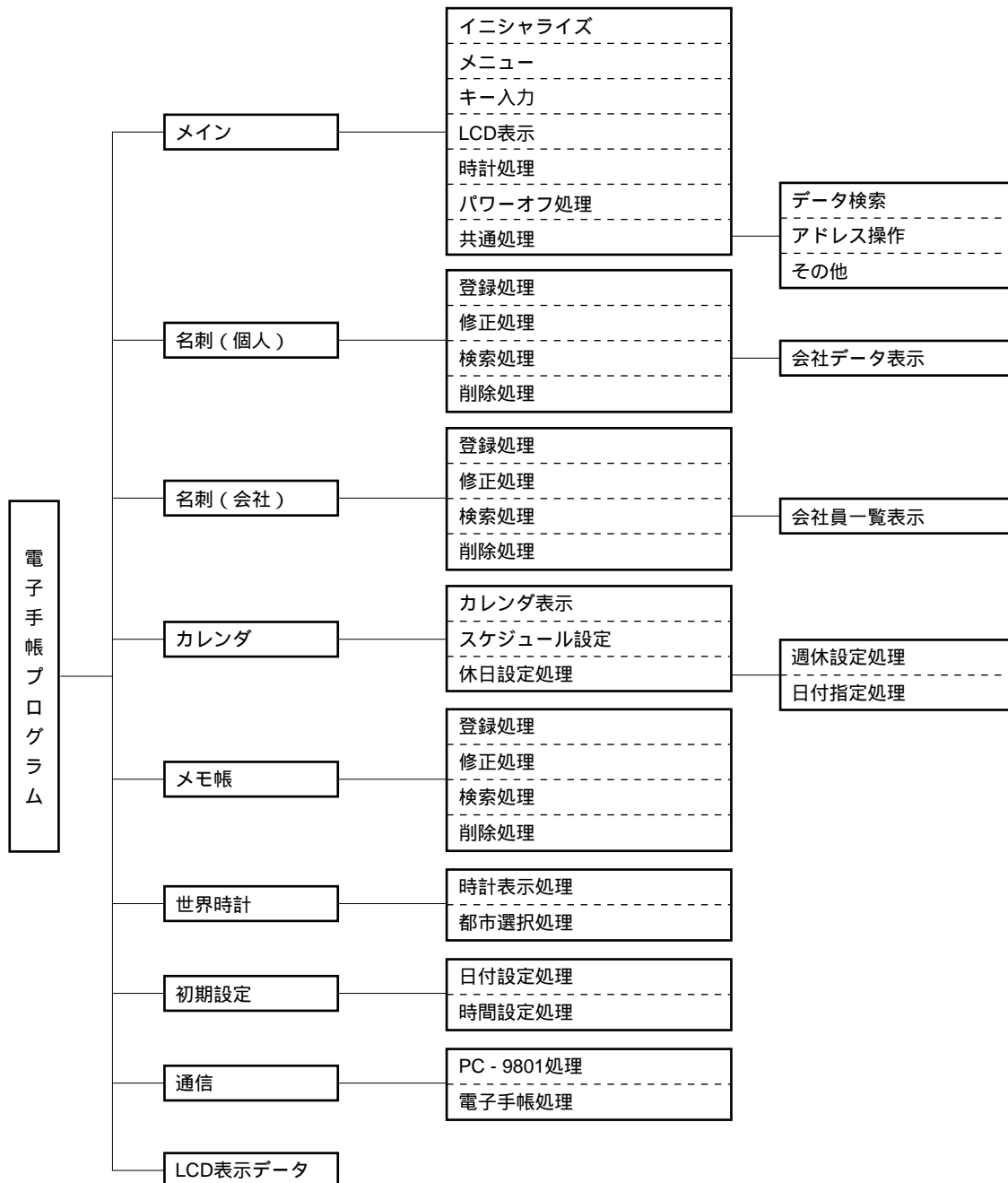
第4章 プログラム説明

4.1 モジュール説明

4.1.1 モジュール構成

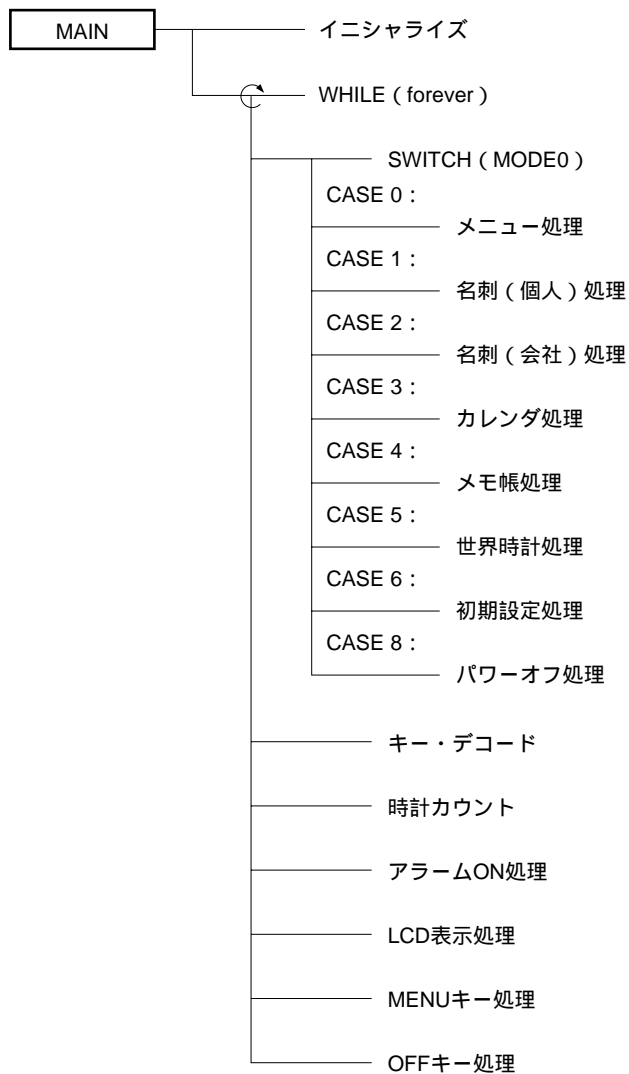
このプログラムのモジュールは、電子手帳の機能ごとに分けた7項目に、メインと表示データ・テーブルの2項目を追加した、9個のモジュールで構成しています。モジュール内容を図4-1に示します。

図4-1 電子手帳モジュール構成図

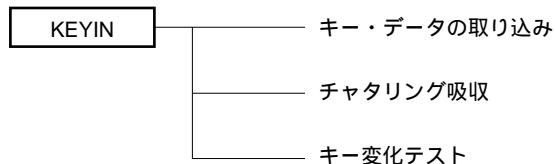


4.1.2 ジェネラルSPDチャート

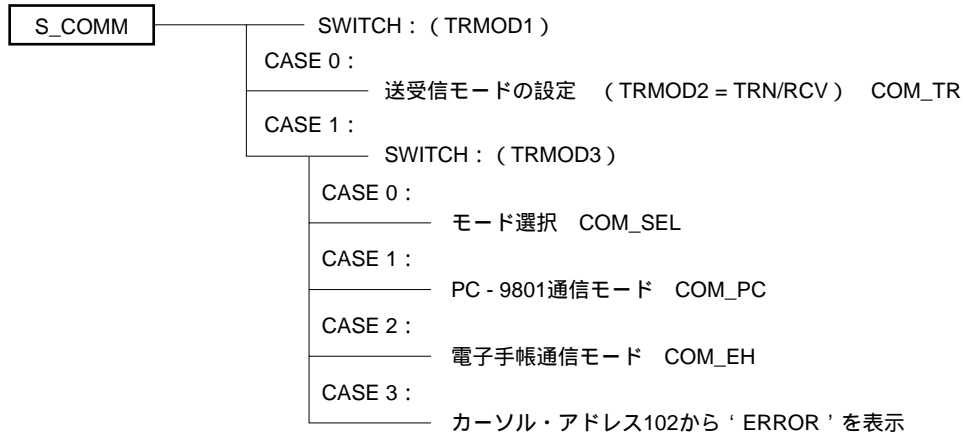
メニュー



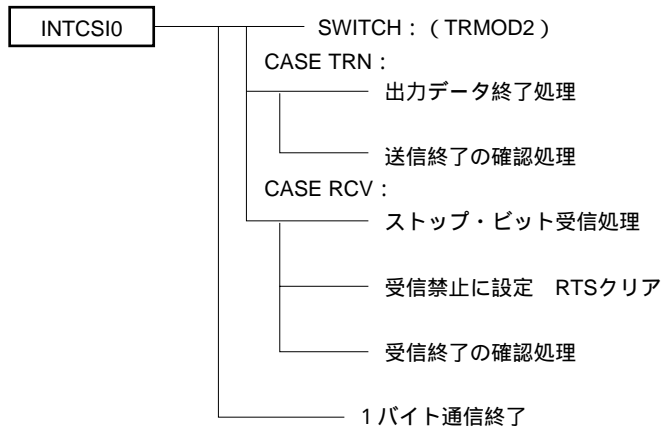
キー入力(8ビット・タイマ1)



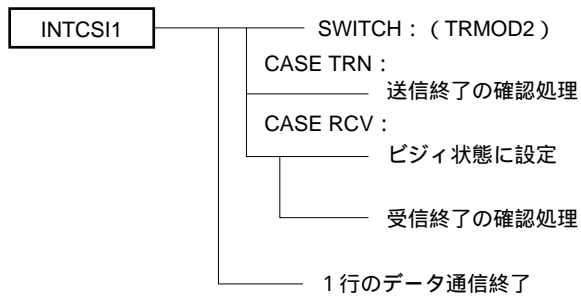
通 信



・ PC-9801



・ 電子手帳同士



4.2 RAM説明

このプログラムで使用するRAMの内容を表4 - 1に、フラグの内容を表4 - 2に示します。

表4 - 1 μ PD78014電子手帳RAM一覧表(1/2)

RAM名称	PUBLIC	領域	モジュール	配置	用途
MODE0		1	MAIN	SADDR	モード選択(ネスティング・レベル1)
MODE1					モード選択(ネスティング・レベル2)
MODE2					モード選択(ネスティング・レベル3)
MODE3					モード選択(ネスティング・レベル4)
MODE4					モード選択(ネスティング・レベル5)
CUR_ADR		8			カーソル・アドレス値
KEY_CD					キー・コード格納
KEYDAT		1			キー・ビット・データ(チャタリング吸収済み)
KEYWK					キー・ビット・データ(チャタリング吸収中)
KEYCHT		1			キー・チャタリング・カウンタ
INV_DS					反転表示の行データ
YEAD					日付年データ
MOND					日付月データ
DAYD		2			日付日データ
HOUD(P)					時計時データ
MIND(P)					時計分データ
SECD(P)		1			時計秒データ
M_BANK					外部メモリ・バンク設定エリア
MEMPAG					メモリ・ページ格納エリア
MEMBNK					メモリ・バンク格納エリア
BACPAG					メモリ・ページのバックアップ
BACBNK					メモリ・バンクのバックアップ
FOR_ADR					検索(前進)メモリ・ページ格納エリア
REV_ADR					検索(後退)メモリ・ページ格納エリア
FOR_BNK					検索(前進)メモリ・バンク格納エリア
REV_BNK				検索(後退)メモリ・バンク格納エリア	
SEACHR				検索文字格納エリア	
i				for文用ワーク・カウンタ	
j		for文用ワーク・カウンタ			
k		キー・スキャン用ワーク・カウンタ			
POF_CTP		2	SADDRP	オート・パワーオフ時間計測カウンタ	
STACK		20H	IHRAM	スタック領域	
LCDDS		240	0FB00H	LCD表示データ設定エリア	
DTWORK		6	0FC00H	データ処理ワーク・エリア	

表4-1 μPD78014電子手帳RAM一覧表(2/2)

RAM名称	PUBLIC	領域	モジュール	配置	用途		
ADM_VOL		1	ADMAN	SADDR	名刺(個人)データ数		
ADC_VOL			ADCOM		名刺(会社)データ数		
PAR_ADR			CALEN		SADDR	カレンダー()表示アドレス	
CE_YEAD						カレンダー表示年データ	
CE_MOND						カレンダー表示月データ	
CE_DAYD						カレンダー表示日データ	
CE_TIME	4	カレンダー検索時間データ					
CE_WEKD		1		SADDR		カレンダー曜日データ	
WEKHOLD			週休データ				
MON_END			指定月最終日格納				
STADAY			指定月1日の曜日格納				
HOL_VOL			休日表示データ数				
MEM_VOL			MEMO		SADDR	メモ帳データ数	
TIT_VOL		メモ帳タイトル(レベル1)データ数					
TIT_PNT		タイトル(レベル1)検索行データ					
FOR_PNT		タイトル(レベル1)検索行(前進)格納					
REV_PNT		タイトル(レベル1)検索行(後退)格納					
TITLE1		10		IHRAM		タイトル(レベル1)を格納	
TITLE2			タイトル(レベル2)を格納				
CIT_DAT		1	WCLOC	SADDR	表示都市データ		
SET_HDT			DTSET		初期設定時データ格納		
SET_MDT					初期設定分データ格納		
SET_SDT					初期設定秒データ格納		
COM_VOL		2	COM	SADDRP	通信データのページ数		
TRNADRP					1	SADDR	通信開始アドレス
TRNADRL							通信開始アドレスのメモリ・バンク
TRN_BNK							モード選択(ネスティング・レベル1)
TRMOD1							モード選択(ネスティング・レベル2)
TRMOD2							モード選択(ネスティング・レベル3)
TRMOD3							送信データ格納
TRNDAT							通信データの表示行数カウンタ
LIN_CT							通信データの文字数カウンタ
CHR_CT							チェック・サム・データ
SUM_DT							受信データ格納アドレス
WORKADR							

表4-2 μPD78014電子手帳フラグー覧表

フラグ名	PUBLIC	モジュール	用	途
KY_SIGF		MAIN	1:有効キーあり	0:有効キーなし
KY_CHGF			1:キー変化あり	0:キー変化なし
KY_STAF			1:キー・チャタリング開始	0:キー・チャタリング中
KY_ENDF			1:キー・チャタリング終了	0:キー・チャタリング中
KY_ONF			1:ONキー確認済み	0:ONキー未確認
KY_OFFF			1:キーOFF状態確認	0:キーOFF状態未確認
KY_ERRF			1:キー・エラー発生	0:キー・データ正常
WT1SF			1:時計0.5秒カウント	0:時計0.5秒カウントなし
SFT_F			1:シフト・キーON	0:シフト・キーOFF
MD_CG0F			1:モード0変更あり	0:モード0変更なし
MD_CG1F			1:モード1変更あり	0:モード1変更なし
MD_CG2F			1:モード2変更あり	0:モード2変更なし
MD_CG3F			1:モード3変更あり	0:モード3変更なし
MD_CG4F			1:モード4変更あり	0:モード4変更なし
IV_CHGF			1:反転表示行変更あり	0:反転表示行変更なし
CR_DSPF			1:カーソル表示あり	0:カーソル表示なし
ALM_ONF			1:アラーム出力あり	0:アラーム出力なし
ALM_ERF			1:アラーム時間一致なし	0:アラーム時間一致あり
STNBYF			1:パワーOFF状態	0:パワーON状態
DS_ERAf			1:LCD表示データ消去	0:-
ADENDF			1:最終アドレス検索終了	0:アドレス検索中
ADSEAF			1:データ登録アドレス検出	0:-
CMPERF			1:データ比較不一致	0:-
LCDCHGF			1:LCD表示データの変更	0:-
CLDCGF		CALEN	1:カレンダー月表示変更	0:-
PAR_1F			1:()位置を1日に設定	0:-
PAR_SET			1:()位置の変更	0:-
TIM_OKF			1:スケジュール時間正常	0:-
COMM_F		COM	1:通信処理中	0:-
TR_CG1F			1:通信モード1変更あり	0:通信モード1変更なし
TR_CG2F			1:通信モード2変更あり	0:通信モード2変更なし
TR_CG3F			1:通信モード3変更あり	0:通信モード3変更なし
COMBYTF			1:1バイト通信中	0:-
COMLINF			1:1行通信中	0:-
TRNSETF			1:送信データ設定済み	0:送信データ未設定
RCVENDF			1:1ページ受信終了	0:-
TR_ENDF			1:通信終了フラグ	0:-
TIM_OKF			DTSET	1:初期設定時間正常

4.3 メイン

メイン・モジュールでは、イニシャライズ、メニュー、パワーオフ、キー入力、LCD表示、時計カウンタの処理、および各モジュールでの共通処理で構成しています。

4.3.1 イニシャライズ

電子手帳のイニシャライズ内容を次に示します。

- ・ 入出力ポートの設定
- ・ タイマ、シリアル・インタフェースの設定
- ・ RAMの初期設定
- ・ 割り込みの設定
- ・ LCDモジュール (EG2402S-AR) の設定
- ・ パワーオフ・モードの設定

4.3.2 メニュー

メニュー処理では、電子手帳のモードをLCD表示し、ENTキー、または1-6のキー入力により、MODE0にデータを書き込みます。

表4-3 メニュー・モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
0	-	-	-	-	メニュー表示

備考 - : don't care

4.3.3 キー入力

キー入力は、8 × 8 キー・マトリクスと、INTP3を使用するONキーで構成しています。8 × 8の取り込みは、8ビット・タイマ1を5msに設定した割り込み処理で行っています。チャタリング吸収は、キー・データの4回一致で見ているため、15-20msになります。キー・スキャン信号は、P30-P32の3本の出力ポートを使用して、デコーダμPD74HC138の制御を行い、8ラインの信号を出力しています。また、キー・リターン信号は、ポート1の8ビットにより行っています。次にキー・マトリクスに関する処理を示します。

取り込みデータは、1キー1ビット対応でメモリに格納しています。

SHIFTキーを除く多重押しは禁止します。

ロック・アウト方式を用いています。

SHIFTキーのオン/オフ用の2種類のテーブルを用意し、キー・デコードします。

文字キーは、表示データにデコードしています。

パワーオン状態のみ有効になります。

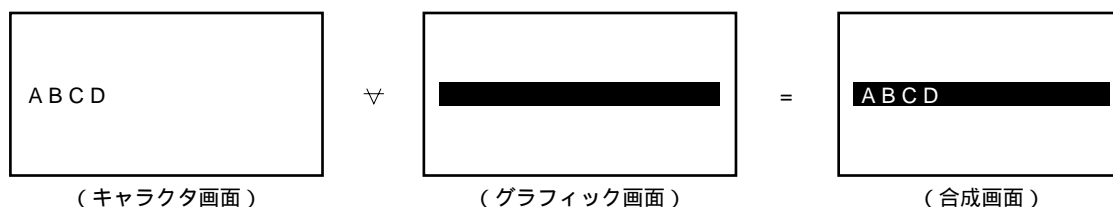
ONキーは、パワーオフ状態（スタンバイ状態）でのみ有効にするため、INTP3のスタンバイ解除信号を使用しています。

4.3.4 LCD表示

LCD表示は、LCDモジュールEG2402S-ARのコントローラSED-1330（セイコーエプソン（株）製）を、μPD78014の外部アクセス領域のFFD0H, FFD1H番地に配置して制御しています。LCDの表示はSED-1330が内蔵しているキャラクタ・ジェネレータを使用して、文字を示します。また、反転表示については、グラフィック画面との排他的理論和により行っています（SED-1330の合成画面コマンドを使用）。

LCDの表示データは、LCD表示用エリア（LCDDS）に設定してあり、表示データに変更があったときのみ、EG2402S-ARにデータを送ります。

図4-2 LCD反転表示方法



4.3.5 時計カウント

時計カウントでは、時計用タイマのカウント・クロックにサブシステム・クロックを選択し、時計用フラグのセット時間を0.5秒に設定しています。したがって、パワーオンのはきは、メイン・ルーチンごとに時計用フラグをテストして時計のカウント・アップを行っています。パワーオフのときは、時計用フラグが1にセットされる（スタンバイ解除信号発生）ごとにスタンバイ・モードを解除して時計のカウント・アップを行っています。カウント・アップは、秒 分 時 日 月 年の順番で行っています。

また、時計カウントの動作を利用して、オート・パワーオフの機能を制御しています。オート・パワーオフは、1秒ごとに16ビットのメモリ(POF_CTP)をカウント・アップし、カウンタが300回になったときに5分間と判断し、パワーオフ・モードに設定します。カウンタのクリアは、キー入力処理でキーに変化があったときに行っています。

4.3.6 パワーオフ処理

パワーオフ処理では、外部デバイスを含みスタンバイ状態に入ります。スタンバイ状態の解除には、時計用タイマとINTP3の2種類の要求信号を使用しています。時計用タイマの解除では、時計のカウント・アップを行い、カウント・アップ終了後にはスタンバイ状態に戻ります。ただし、スケジュール・モードで設定したアラームONのデータと一致したときは、メニュー・モードになります。INTP3の解除では、ONキーの入力ありと判断し、メニュー・モードになります。

表4 - 4 パワーオフ・モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
8	-	-	-	-	メニュー表示

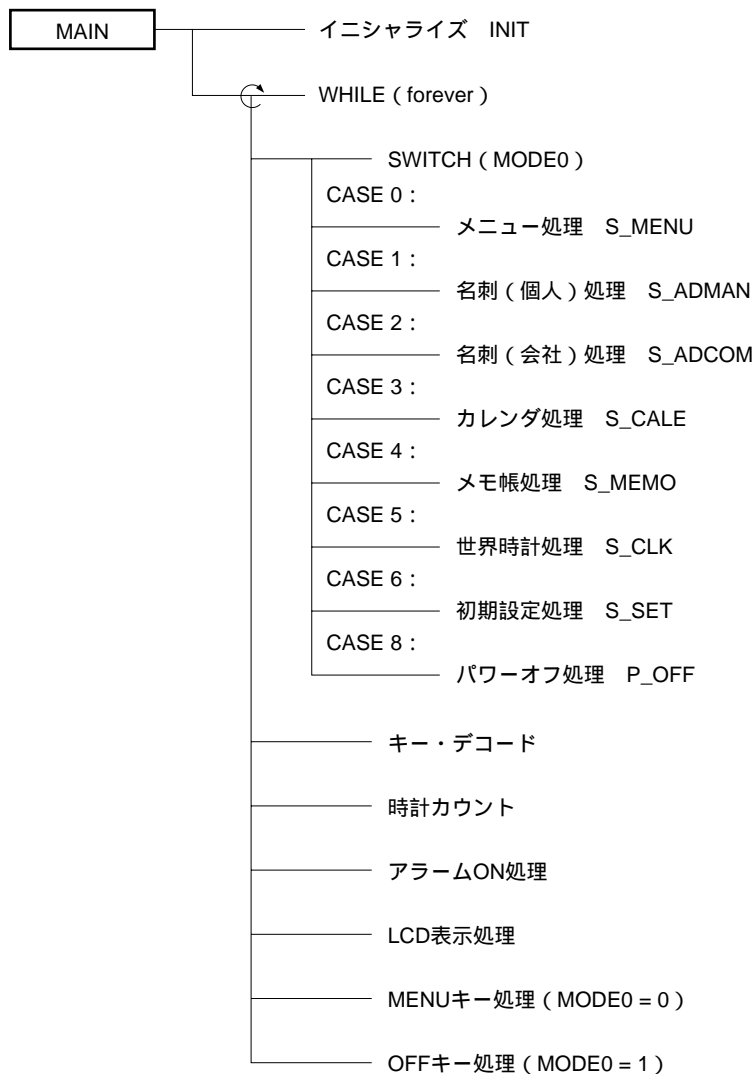
備考 - : don't care

4.3.7 共通処理

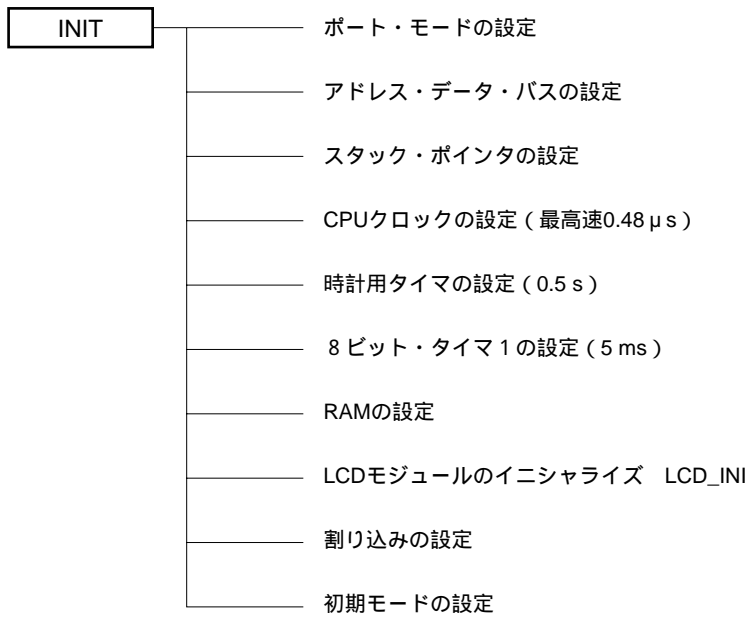
メイン・モジュールでは、4.3.1 - 4.3.6以外の処理にデータの検索、登録、削除や外部メモリのアドレス操作などがあります。

4.3.8 SPDチャート

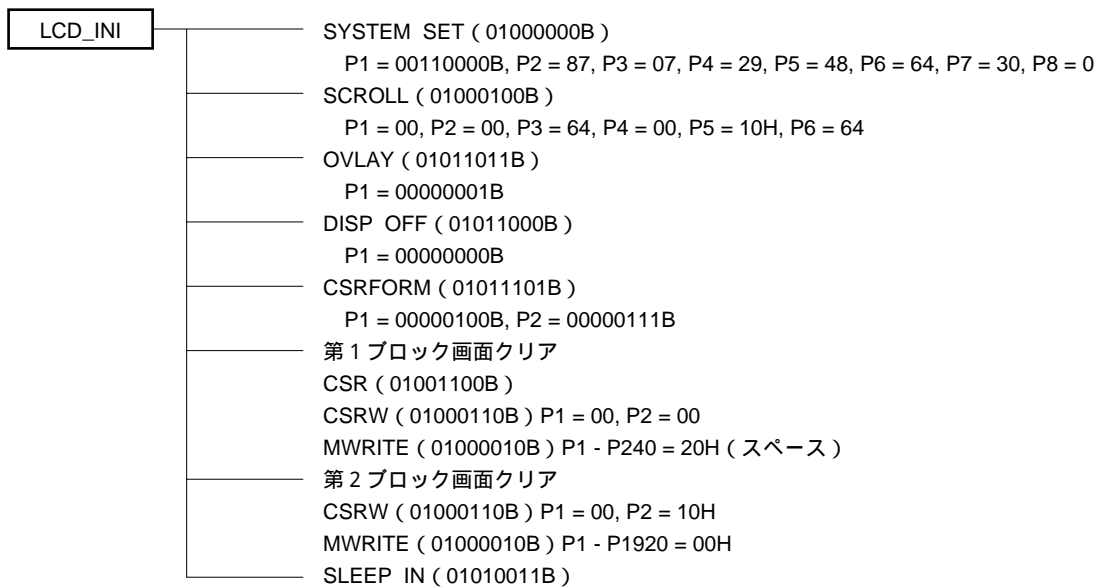
・メニュー



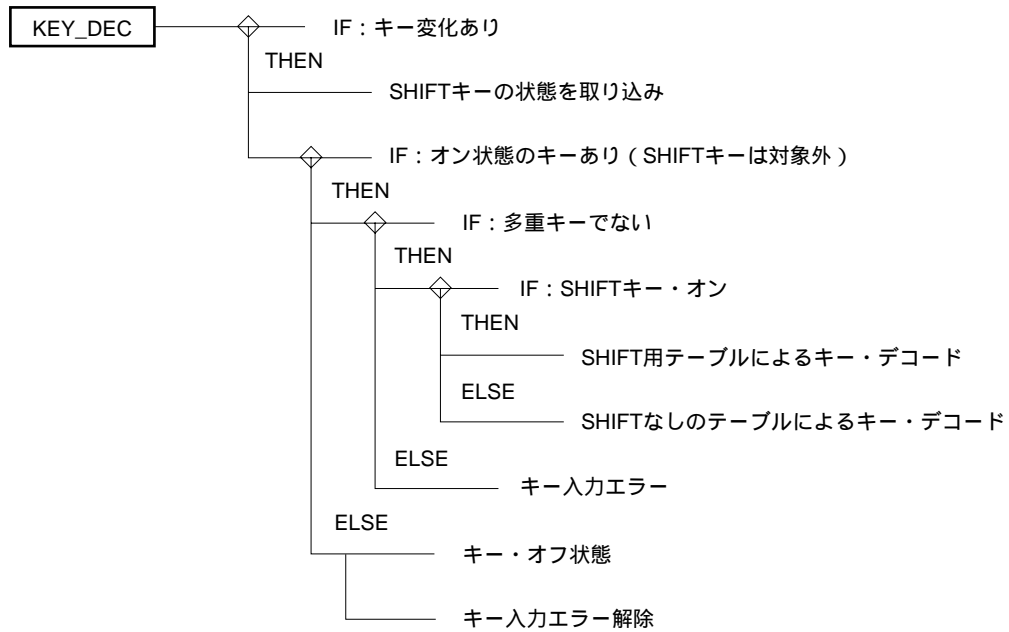
・イニシャライズ



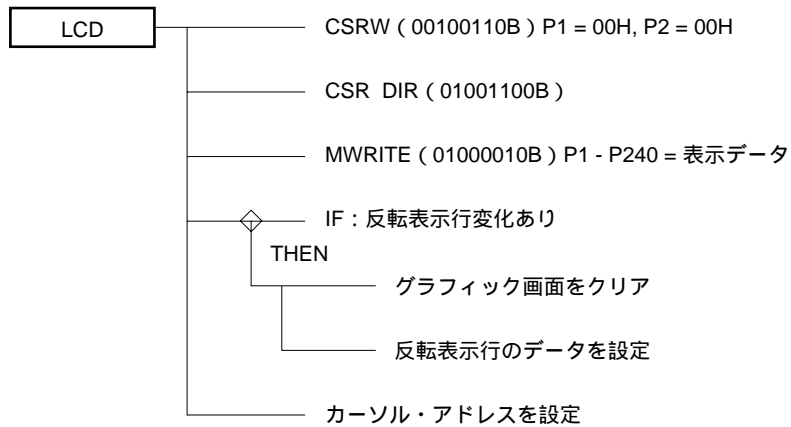
・LCDモジュールのイニシャライズ



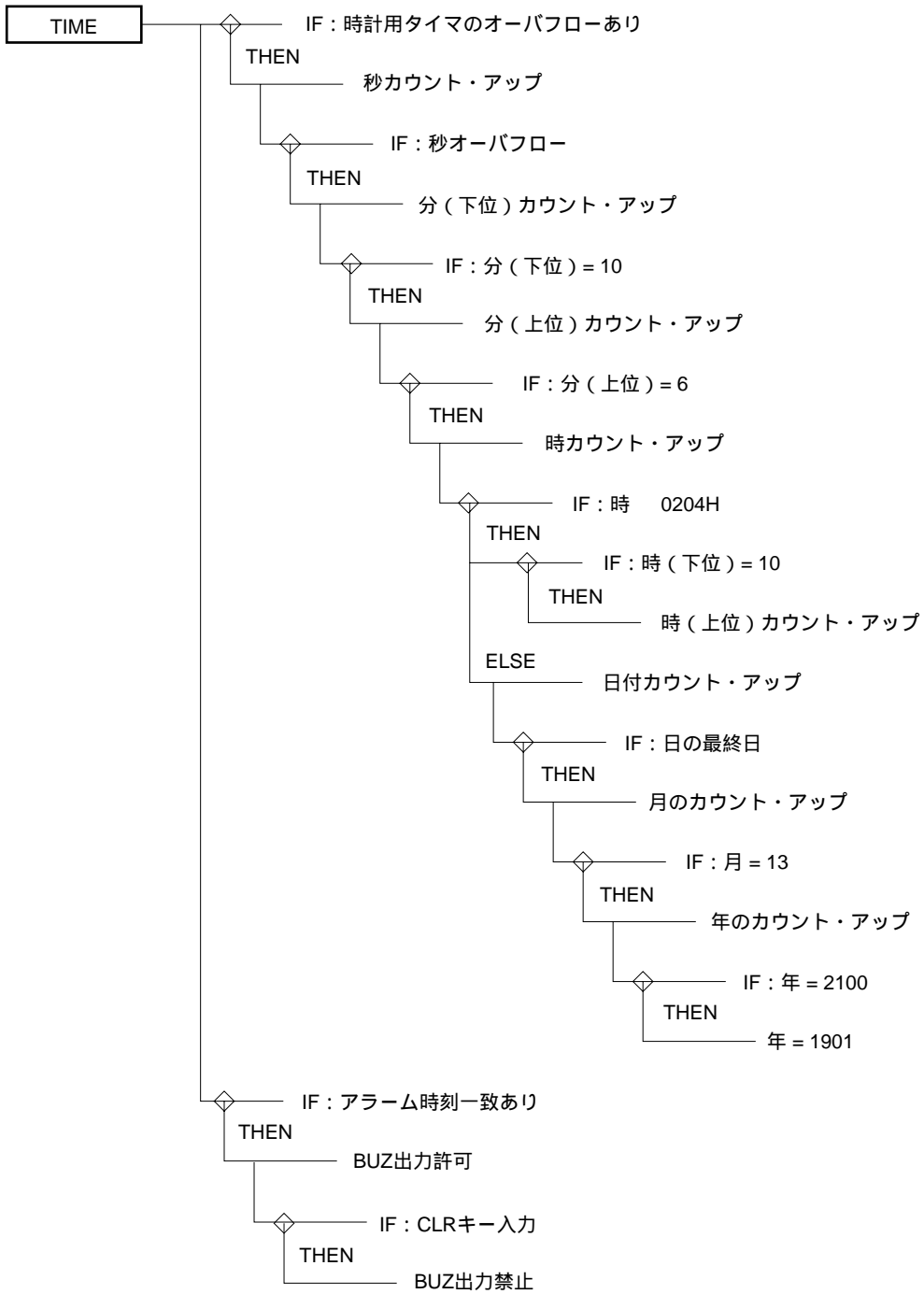
・キー・デコード



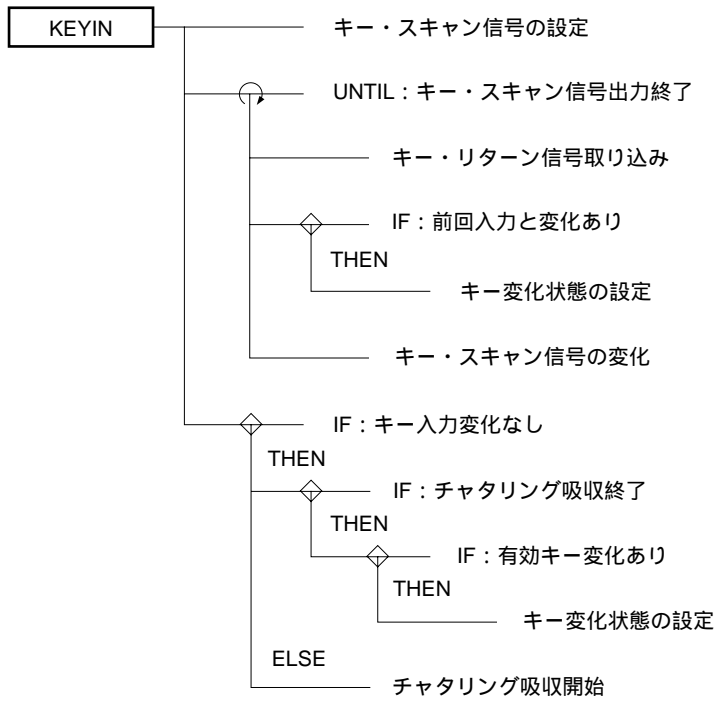
・LCD表示



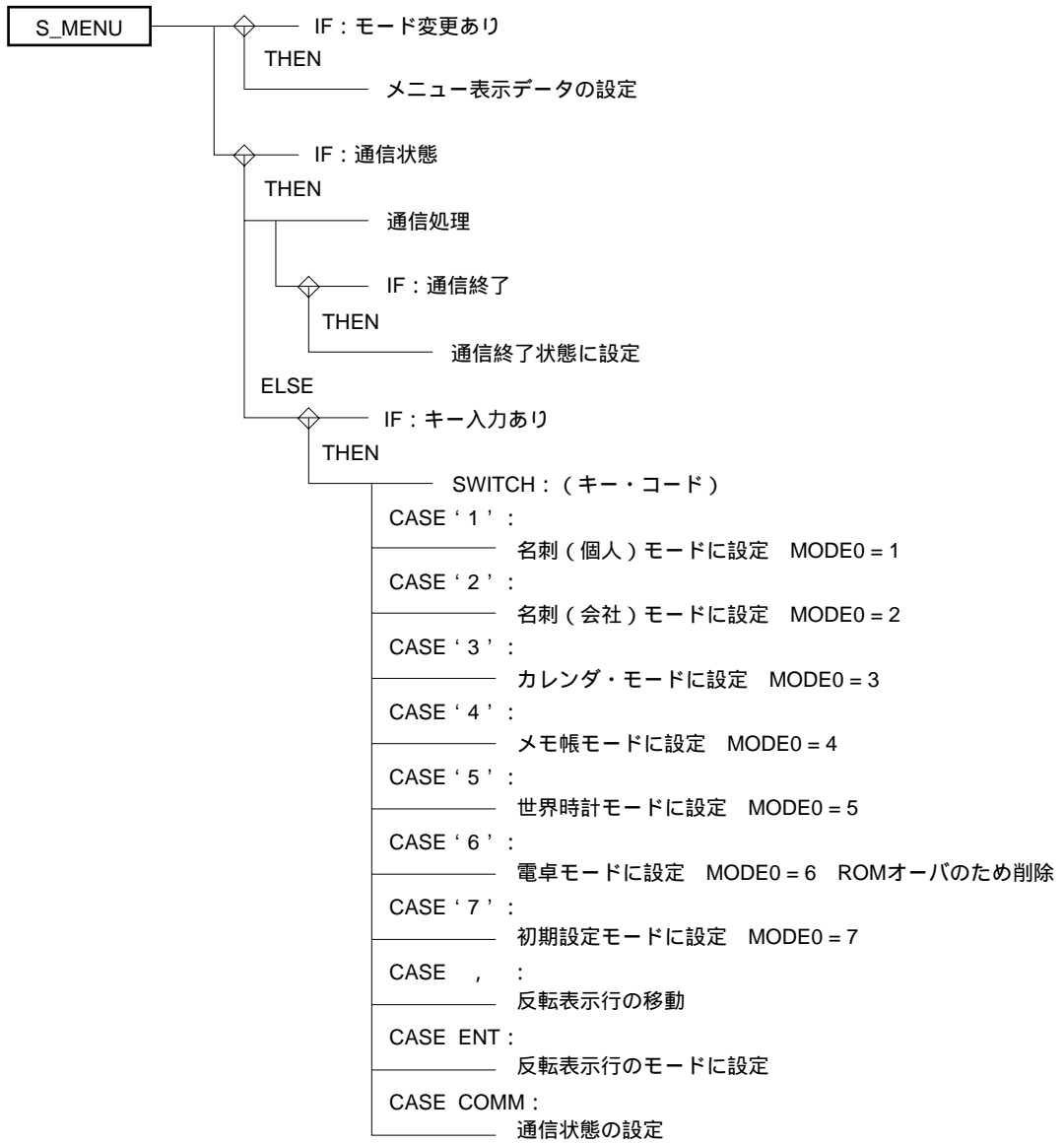
・時計カウント



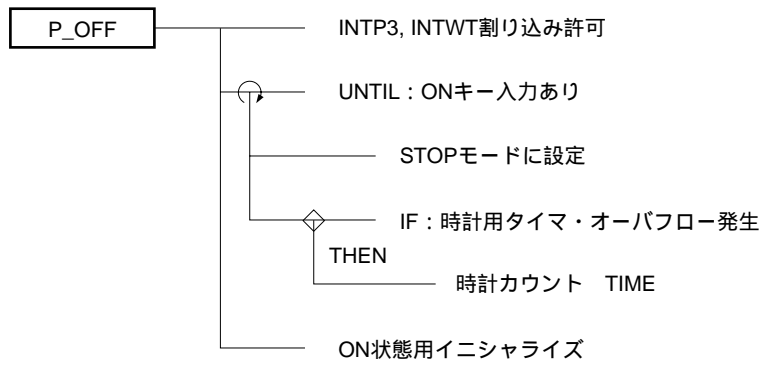
・キー入力



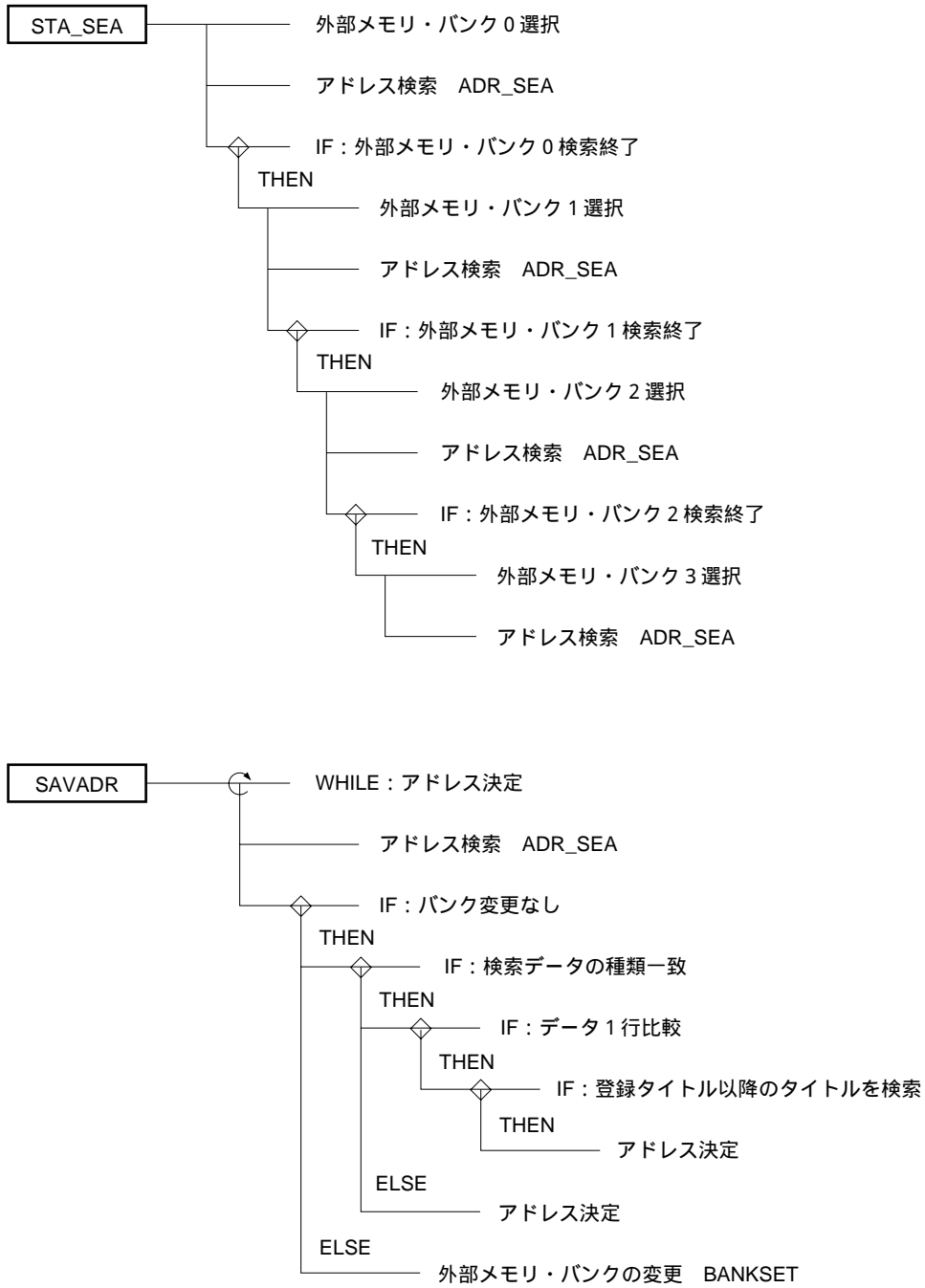
・メニュー



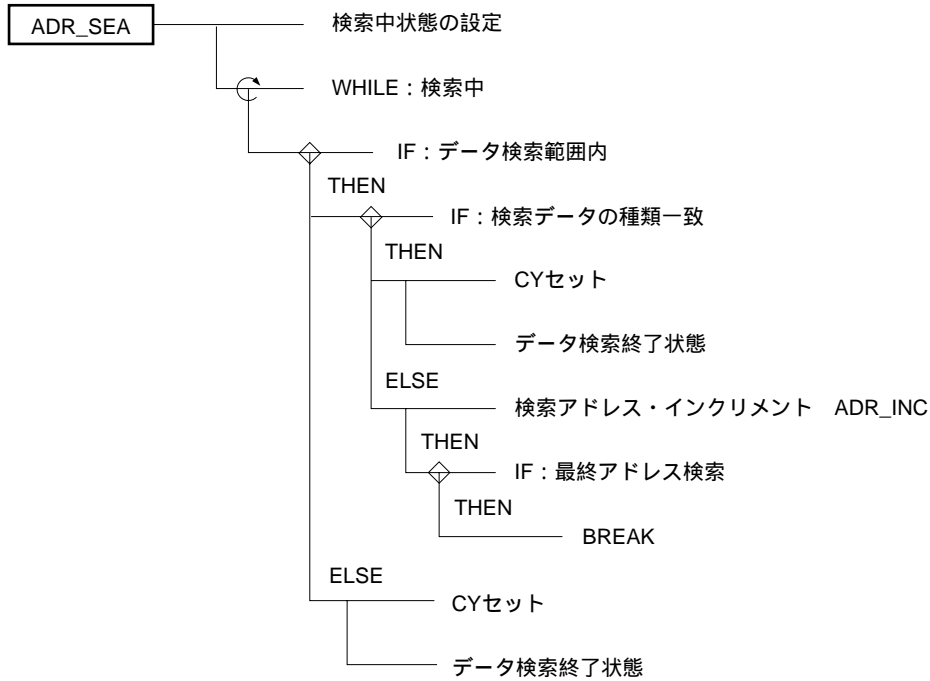
・パワーオフ処理



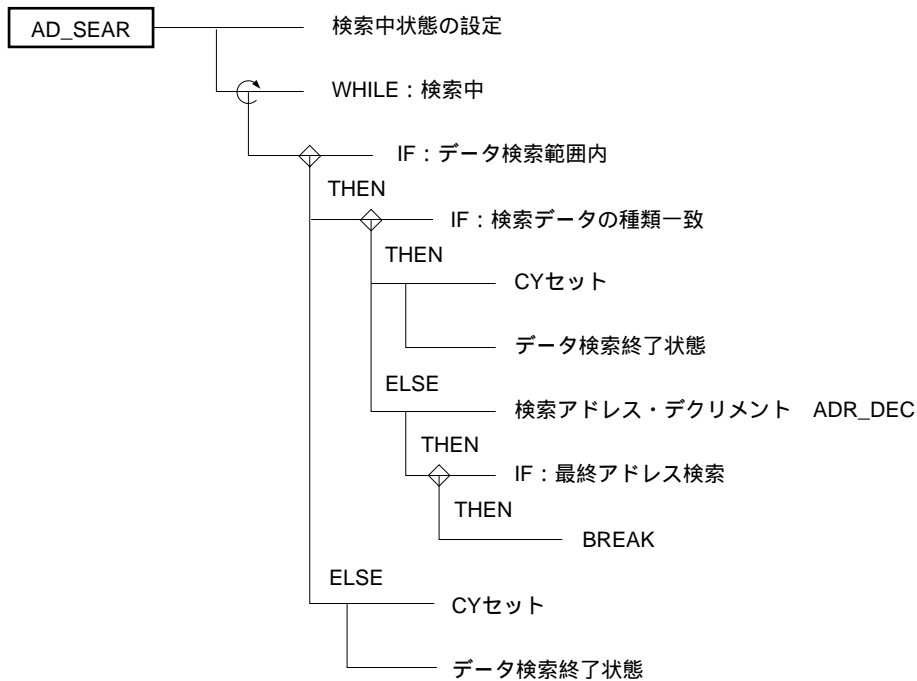
・データ検索



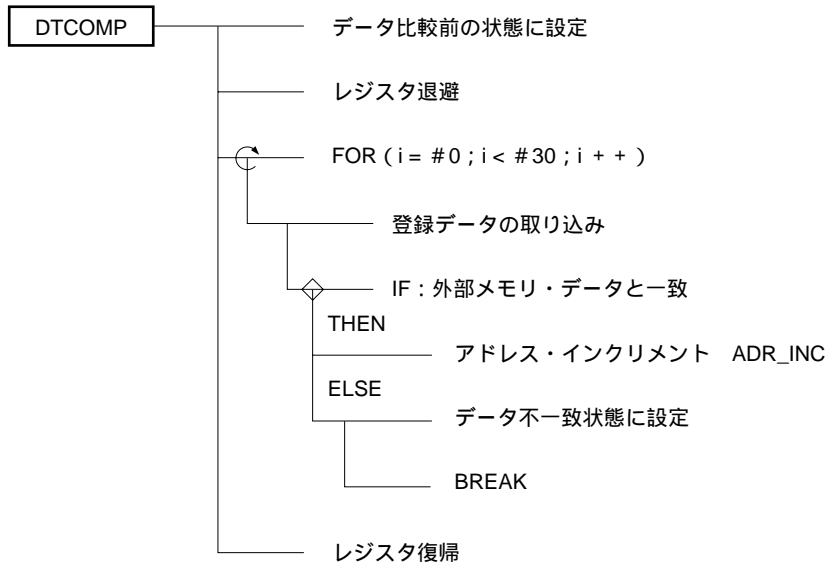
・データ・アドレス検索（前進）



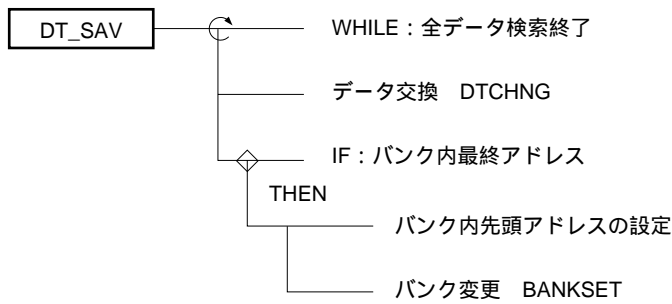
・データ・アドレス検索（後退）



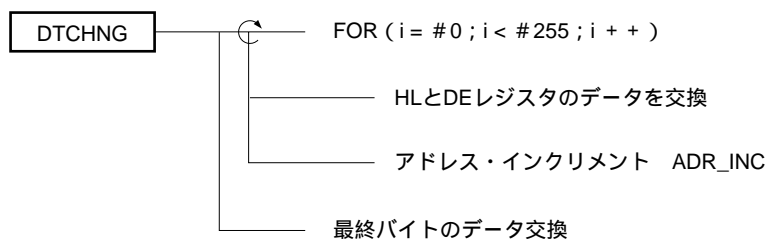
・データ1ライン比較



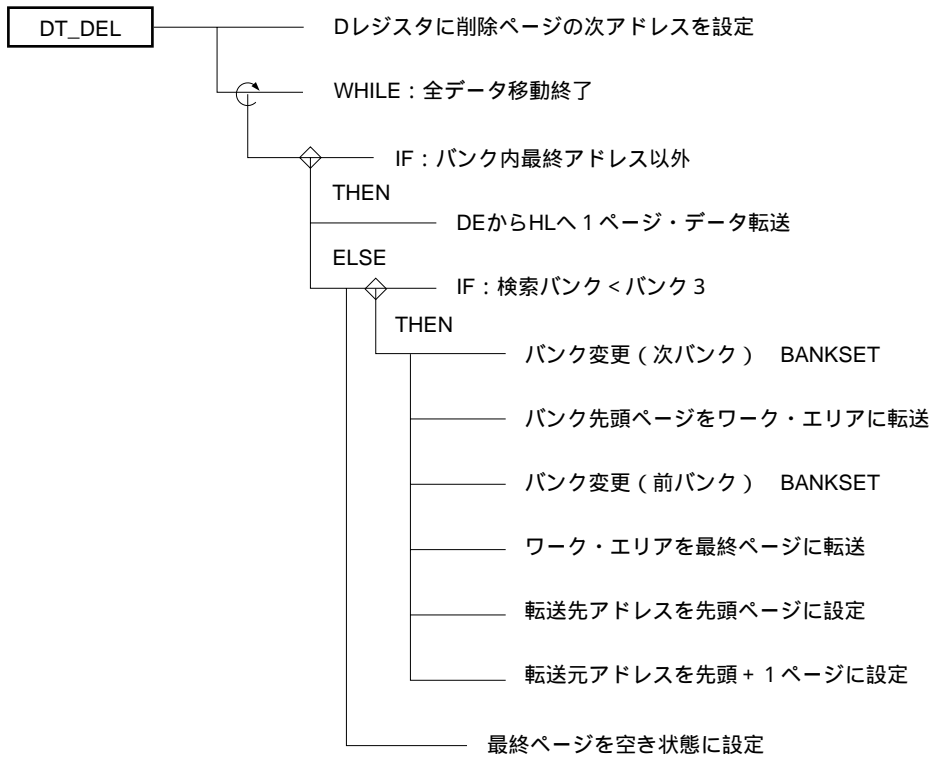
・データ登録



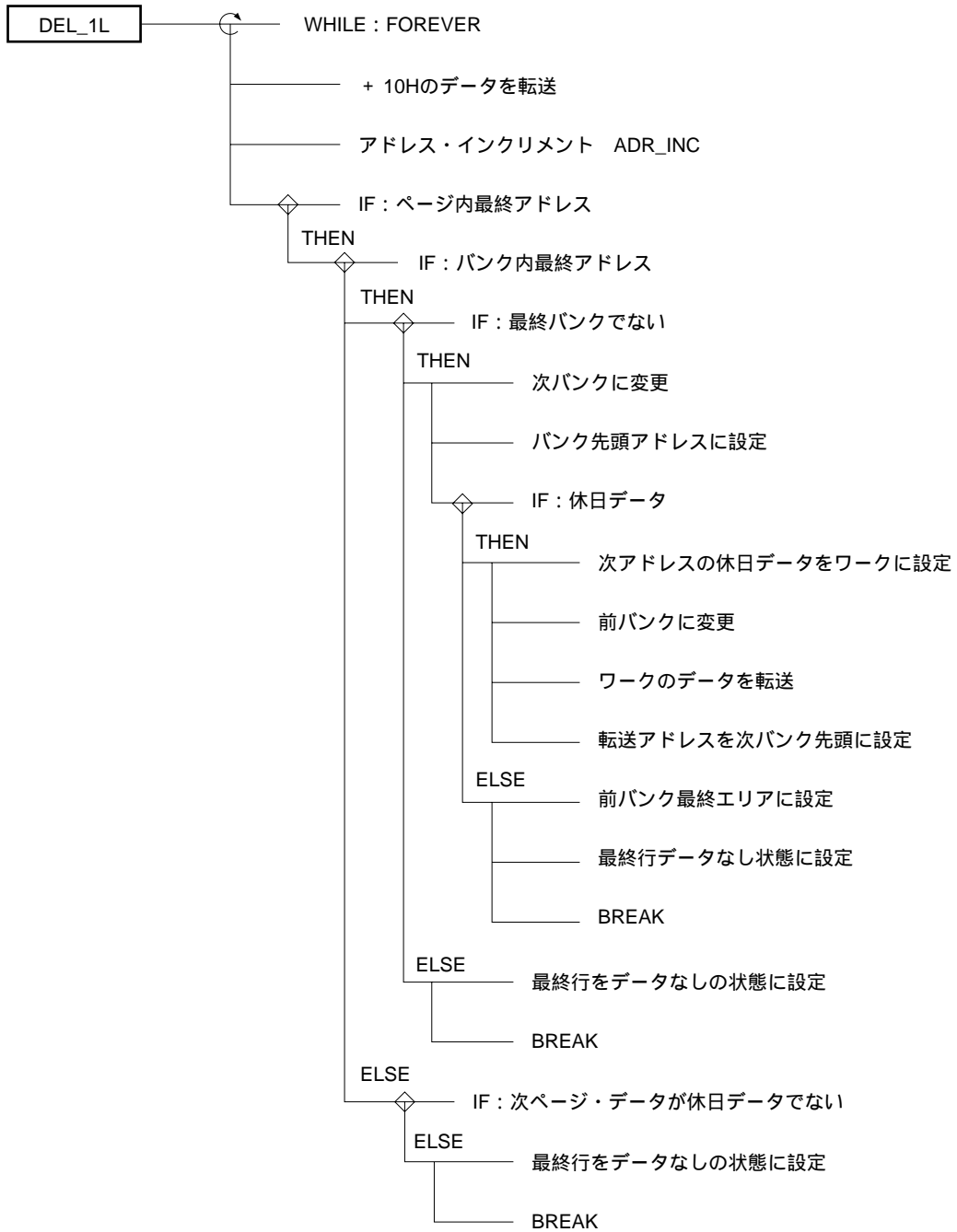
・データ交換



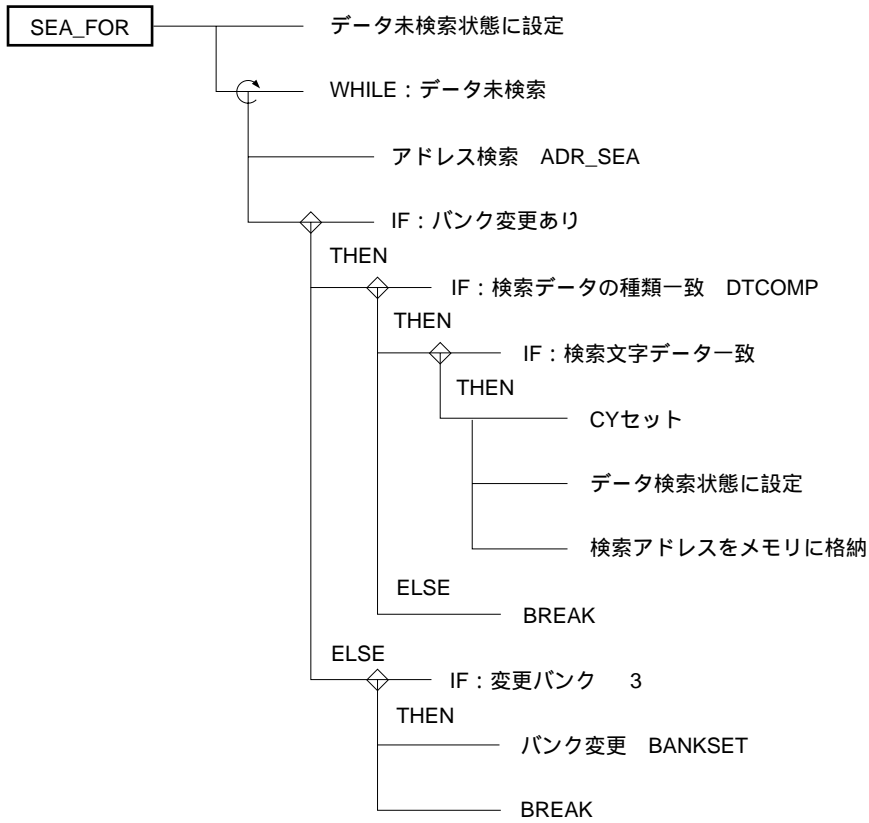
・データ削除



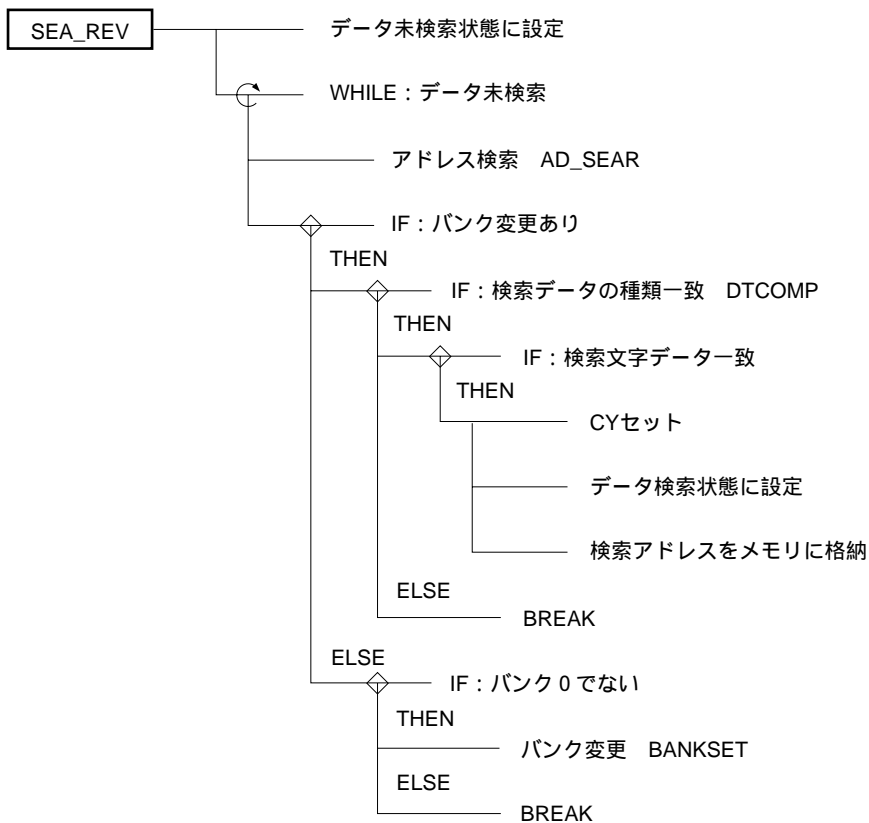
・1ライン・データ削除



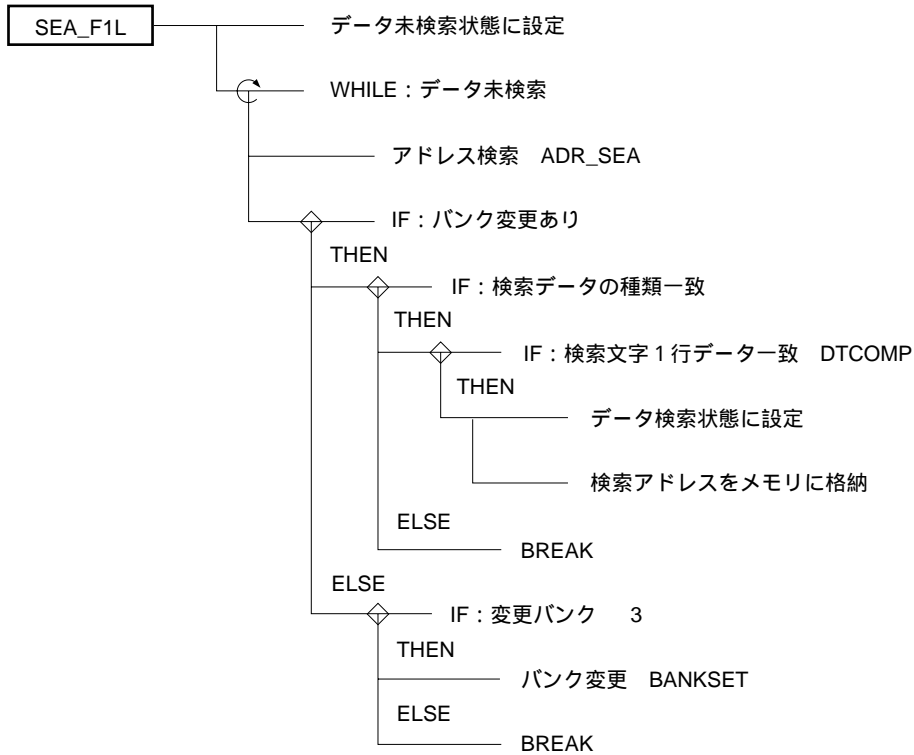
・データ検索（前進）



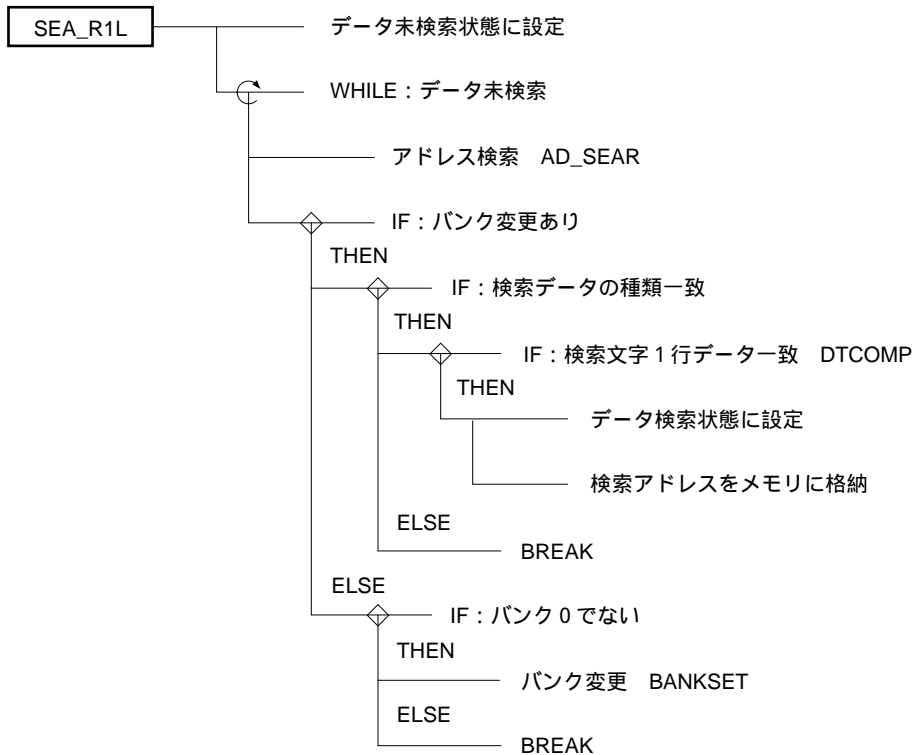
・データ検索（後退）



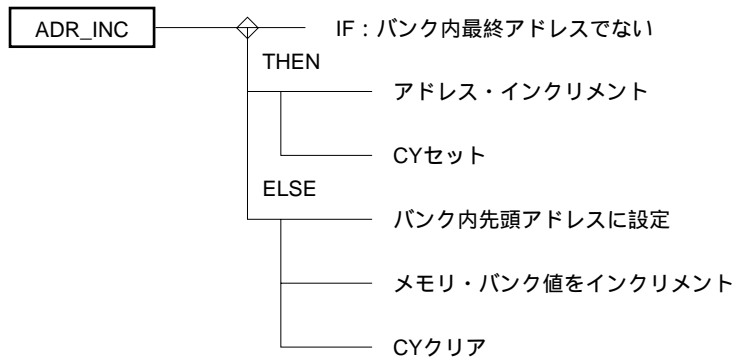
・ 1行データ検索 (前進)



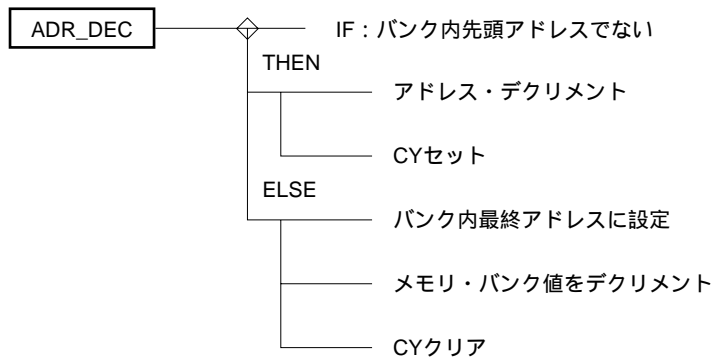
・ 1行データ検索 (後退)



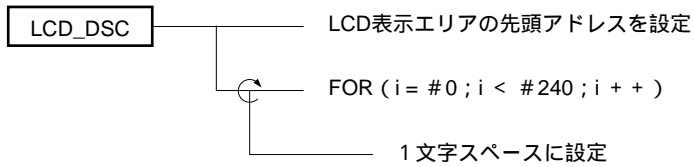
・外部メモリのアドレス・インクリメント



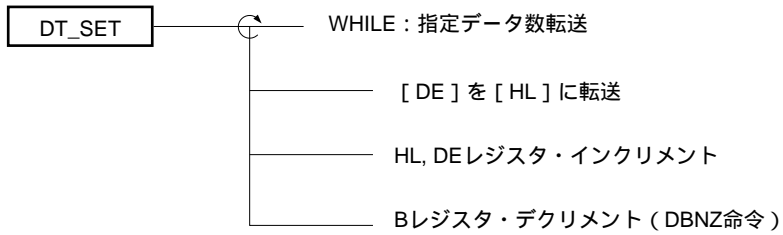
・外部メモリのアドレス・デクリメント



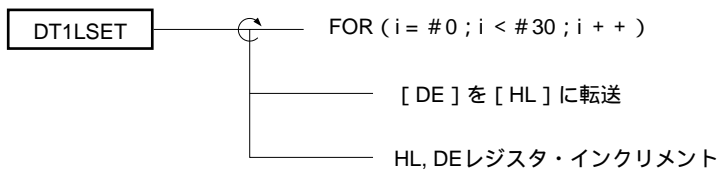
・LCD表示エリアのクリア



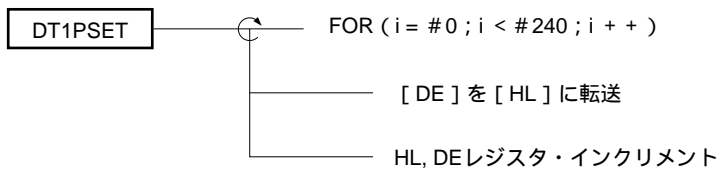
・メモリ間のデータ転送



・LCD 1行のデータ・セット



・LCD 1ページのデータ・セット



4.3.9 プログラム・リスト

\$PC(014)

\$NOLIST

```

PUBLIC  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
PUBLIC  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
PUBLIC  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
PUBLIC  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
PUBLIC  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
PUBLIC  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, MEMPAG, MEBNKB, BACPAG, BACBNK
PUBLIC  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
PUBLIC  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, LCDCHGF
PUBLIC  IV_CHGF, DS_ERAF, CR_DSPF, ADENDF, ADSEAF, CMPERF
PUBLIC  VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT, TIT_DAT
PUBLIC  DTWORK, LCDDS, LCDCLR, LCD_DSC, DTILSET, DTIPSET, LCD_C, LCD_D, DT_SAV, STA_SEA
PUBLIC  SAVADR, DTCOMP, DT_DEL, DEL_IL, BANKSET, ADR_SEA, AD_SEAR, ADR_INC, ADR_DEC, DT_SET
PUBLIC  SEA_FOR, SEA_REV, SEA_FIL, SEA_RIL, ASC_DEL, NEX_ADR, BEF_ADR
PUBLIC  ADRSTA, ADREND

EXTRN  S_ADMAN, S_ADCOM, S_CALE, S_MEMO, S_CLK, S_SET, S_COMM      ;S_CALC
EXTRN  CAL_LEA, CAL_NOR, ADM_VOL, ADC_VOL, MEM_VOL, COM_VOL, TRN_ADR, TRN_BNK, ALMON, ALMOFF, CIT_DAT
EXTRN  COMM_F, TR_CG1F, TR_CG2F, TR_CG3F

```

\$LIST

```

;*****
;          ベクタ・テーブル定義
;*****
VERES  CSEG      AT 00H
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
VECT_N1 CSEG      AT 08H
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
VECT_N2 CSEG      AT 12H
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
VETIM1 CSEG      AT 16H
      DW      INTTM1           ;8ビット・タイマ1用ベクタ・テーブルの設定
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)
VENT_N3 CSEG      AT 3EH
      DW      RES_STA          ;リセット用ベクタ・テーブルの設定 (未使用)

$      INCLUDE (PORT.INC)
$      INCLUDE (LCDCMD.INC)

;*****
;          キー・コード定義
;*****
CUR_UP EQU      00H          ;↑
CUR_DN EQU      01H          ;↓
CUR_LE EQU      02H          ;←
CUR_RI EQU      03H          ;→
ENTER  EQU      04H          ;ENT
DELETE EQU      05H          ;DEL
CLEAR  EQU      06H          ;CLR
UP     EQU      07H          ;UP
DOWN  EQU      08H          ;DOWN
MENU  EQU      09H          ;MENU
COMM  EQU      0AH          ;COMM
OFF   EQU      0BH          ;OFF
MR    EQU      0CH          ;MR
MC    EQU      0DH          ;MC
M_ADD EQU      0EH          ;M+
M_SUB EQU      0FH          ;M-
S_CURU EQU      10H         ;SHIFT+↑
S_CURD EQU      11H         ;SHIFT+↓
S_CURL EQU      12H         ;SHIFT+←
S_CURR EQU      13H         ;SHIFT+→
SIG_INV EQU      14H        ;+/-
S_DEL  EQU      15H         ;SHIFT+DEL

```




```

S_UP EQU 17H ;SHIFT+U P
S_DOWN EQU 18H ;SHIFT+D OWN
SPC EQU 20H ;スペース
PAR_LE EQU 28H ;(
PAR_RI EQU 29H ;)
MUL_SIG EQU 2AH ;*
ADD_SIG EQU 2BH ;+
SUB_SIG EQU 2DH ;-
POINT EQU 2EH ;.
DIV_SIG EQU 2FH ;/
COLON EQU 3AH ;:
EQU_SIG EQU 3DH ;=
QUEST EQU 3FH ;?

;*****
; 外部データ・コード定義
;*****
VAC_DAT EQU 0FFH ;空き領域表示データ
ADM_DAT EQU 01 ;名刺(個人)表示データ
ADC_DAT EQU 02 ;名刺(会社)表示データ
MEM_DAT EQU 03 ;メモ帳表示データ
SCA_DAT EQU 04 ;スケジュール表示データ
HOL_DAT EQU 05 ;休日表示データ
TIT_DAT EQU 06 ;メモ帳タイトル1表示データ

;*****
; R A M定義
;*****

DSEG SADDR
MODE0: DS 1 ;モード選択(ネスティング・レベル1)
MODE1: DS 1 ;モード選択(ネスティング・レベル2)
MODE2: DS 1 ;モード選択(ネスティング・レベル3)
MODE3: DS 1 ;モード選択(ネスティング・レベル4)
MODE4: DS 1 ;モード選択(ネスティング・レベル5)
CUR_ADR: DS 1 ;カーソル・アドレス値
KEY_CD: DS 1 ;キー・コード格納
KEYDAT: DS 8 ;キー・ビット・データ(チャタリング吸収済み)
KEYWK: DS 8 ;キー・ビット・データ(チャタリング吸収中)
KEYCHT: DS 1 ;キー・チャタリング・カウンタ
INV_DS: DS 1 ;反転表示の行データ
YEAD: DS 1 ;日付年データ
MOND: DS 1 ;日付月データ
DAYD: DS 1 ;日付日データ
HOUD: ;
HOUDP: DS 2 ;時計時データ
MIND: ;
MINDP: DS 2 ;時計分データ
SECD: ;
SECDP: DS 2 ;時計秒データ
M_BANK: DS 1 ;外部メモリ・バンク設定エリア
MEMPAG: DS 1 ;メモリ・ページ格納エリア
MEMBNK: DS 1 ;メモリ・バンク格納エリア
BACPAG: DS 1 ;メモリ・ページのバックアップ
BACBNK: DS 1 ;メモリ・バンクのバックアップ
FOR_ADR: DS 1 ;検索(前進)メモリ・ページ格納エリア
REV_ADR: DS 1 ;検索(後退)メモリ・ページ格納エリア
FOR_BNK: DS 1 ;検索(前進)メモリ・バンク格納エリア
REV_BNK: DS 1 ;検索(後退)メモリ・バンク格納エリア
SEACHR: DS 1 ;検索文字格納エリア
i: DS 1 ;f o r文用ワーク・カウンタ
j: DS 1 ;f o r文用ワーク・カウンタ
k: DS 1 ;キー・スキャン用ワーク・カウンタ

DSEG SADDR
POF_CTP: DS 2 ;オートパワーオフ時間計測カウンタ

DSPDAT DSEG AT 0FB00H
LCDDS: DS 240 ;LCD表示データ設定エリア

WORK DSEG AT 0FC00H
DIWORK: DS 256 ;データ処理ワーク・エリア
    
```

```

BSEG
KY_SIGF DBIT      ;キー有効フラグ
KY_CHGF DBIT      ;キー変化フラグ
KY_STAF DBIT      ;キー・チャタリング開始フラグ
KY_ENDF DBIT      ;キー・チャタリング終了フラグ
KY_ONF  DBIT      ;キー・オン・フラグ
KY_OFFF DBIT      ;キー・オフ・フラグ
KY_ERRF DBIT      ;キー・エラー・フラグ
WT1SF  DBIT      ;時計1秒カウント
SFT_F   DBIT      ;シフト・キー・フラグ
MD_CG0F DBIT      ;モード0変更フラグ
MD_CG1F DBIT      ;モード1変更フラグ
MD_CG2F DBIT      ;モード2変更フラグ
MD_CG3F DBIT      ;モード3変更フラグ
MD_CG4F DBIT      ;モード4変更フラグ
IV_CHGF DBIT      ;反転表示行変更フラグ
CR_DSPF DBIT      ;カーソル表示フラグ
ALM_ONF DBIT      ;アラーム出力フラグ
STNBYF DBIT      ;スタンバイ設定フラグ
DS_ERAF DBIT      ;表示データ消去フラグ
ADENDF DBIT      ;最終アドレス検索終了フラグ
ADSEAF DBIT      ;データ登録アドレス検出フラグ
CMPERF DBIT      ;データ比較不一致検出フラグ

LCDCHGF DBIT      ;LCD表示データの変更

ST_DAT  DSEG      AT 0FD00H
DS       DS        20H      ;スタック領域

STACK:

LCD_D   EQU      0FFD0H      ;LCD C/Dデータ格納
LCD_C   EQU      0FFD1H      ;LCD C/Dコマンド格納

SFTKEY  EQU      (KEYDAT+7).7 ;SHIFTキー対応ビット
NOKEY   EQU      0FFH      ;キー・オフ状態のキー・コード
ADRSTA  EQU      80H      ;スタート・アドレスを設定
ADREND  EQU      0BFH      ;外部メモリ最終アドレス
AT_OFCT EQU      300      ;5分=300秒(1秒1カウント)

M_SEG   CSEG

;*****
;      インシャライズ
;*****
RES_STA:
    SP=#STACK
    SEL    RBO
    MM=#00000111B      ;フル・アドレス・モード
    P2=#10000100B      ;SCK0/P27, SCK1/P22の出力ラッチをセット
    PM2=#00101001B      ;シリアル/ビジィを出力モードに設定
    P3=#01100000B
    PM3=#10001000B
    PU0=#00000010B      ;ポート1プルアップ抵抗接続
    P6=#00000000B
    PM6=#11111100B

    PCC=#00000000B      ;CPUクロックを最高速に設定
    TCL2=#11010000B      ;BUZ出力2kHzに設定
    TMC2=#00000110B      ;時計用タイマ0.5秒動作モード
    CR10=#163
    TCL1=#01101100B      ;タイマ1のインターバルを5msに設定
    TMC1=#00000001B      ;タイマ1, 2のカウント・クロック (fx/2^8, fx/2^2)を設定
    ;8ビット・タイマ1動作許可

    HL=#0FB00H      ;内部RAMクリア
    BC=#0FEE0H-0FB00H ;RAMデータ数設定
    if(C1=#0) (A)
        B++
    endif

    A=#00H
CLRLOOP:      ;RAMクリア処理中
    [HL]=A      ;RAMクリア
    HL++        ;アドレス・インクリメント
    DBNZ    C, $CLRLOOP ;設定データ数デクリメント

```

```

DBNZ B, $CLRLOOP

M_BANK=#0 ;バンク 0 設定
CALL !BANKSET
HL=#8000H ;外部RAMバンク 0 クリア
BC=#0C000H-8000H ;RAMデータ数設定
if(CI=#0) (A)
  B++
endif
A=#VAC_DAT
CALL !RAMCLR

M_BANK=#1 ;バンク 1 設定
CALL !BANKSET
HL=#8000H ;外部RAMバンク 1 クリア
BC=#0C000H-8000H ;RAMデータ数設定
if(CI=#0) (A)
  B++
endif
A=#VAC_DAT
CALL !RAMCLR

M_BANK=#2 ;バンク 2 設定
CALL !BANKSET
HL=#8000H ;外部RAMバンク 2 クリア
BC=#0C000H-8000H ;RAMデータ数設定
if(CI=#0) (A)
  B++
endif
A=#VAC_DAT
CALL !RAMCLR

M_BANK=#3 ;バンク 3 設定
CALL !BANKSET
HL=#8000H ;外部RAMバンク 3 クリア
BC=#0C000H-8000H ;RAMデータ数設定
if(CI=#0) (A)
  B++
endif
A=#VAC_DAT
CALL !RAMCLR

YEAD=#92 ;カレンダー初期設定 1 9 9 2 / 7 / 1
MOND=#7
DAYD=#1
CIT_DAT=#1 ;世界時計初期データ (NEW YORK)
CALL !LCD_INIT ;LCDイニシャライズ
CLR1 PPRO ;INT P 0 高優先レベルに設定
CLR1 CSIPRO ;INT CS I 0 高優先レベルに設定
CLR1 TMMK1 ;8ビット・タイム1 割り込み許可
EI
MODE0=#8 ;パワーオフ・モードに設定

;*****
; ;メイン処理
;*****
while(forever)
  switch(MODE0)
    case 0:
      CALL !S_MENU ;メニュー処理
      break
    case 1:
      CALL !S_ADMAN ;名刺(個人)処理
      break
    case 2:
      CALL !S_ADCOM ;名刺(会社)処理
      break
    case 3:
      CALL !S_CALE ;カレンダー処理
      break
    case 4:
      CALL !S_MEMO ;メモ帳処理
      break
    case 5:
      CALL !S_CLK ;世界時計処理

```

```

        break
case 6:
    CALL  !S_SET           ;初期設定処理
    break
:
:
:
case 7:
    CALL  !S_CALC         ;電卓処理
    break
:
:
case 8:
    CALL  !P_OFF          ;パワーオフ処理
ends

CALL  ! TIME             ;時計カウント処理
CALL  !LCD               ;LCD表示処理
CALL  !KEY_DEC           ;キー・デコード処理
if_bit(KY_SIGF)         ;有効キーあり
    if(KEY_CD==#MENU)   ;MENUキー・オン?
        CLR1  KY_SIGF
        CLR1  CR_DSPF
        MODE0=#0       ;処理モードをメニュー・モードに設定
        SET1  MD_CGOF
        if_bit(COMM_F)
            CLR1  COMM_F           ;通信状態解除
            SET1  RS_STN         ;μPD4711スタンバイに設定
            SET1  CSINKO        ;通信使用割り込み禁止
            SET1  CSINK1
            SET1  PMKO
        endif
        elseif(KEY_CD==#OFF)     ;OFFキー・オン
            CLR1  KY_SIGF
            MODE0=#8             ;処理モードをパワーオフに設定
            SET1  MD_CGOF
            MODE1=#0
            MODE2=#0
            MODE3=#0
            MODE4=#0
            if_bit(COMM_F)
                CLR1  COMM_F           ;通信状態解除
                SET1  RS_STN         ;μPD4711スタンバイに設定
                SET1  CSINKO        ;通信使用割り込み禁止
                SET1  CSINK1
                SET1  PMKO
            endif
        endif
    endif
endif
endw

;*****
;      メモリ・バンク設定処理
;*****
BANKSET:
    switch(M_BANK)
case 0:
    P6#=#11111100B           ;バンク0設定
    break
case 1:
    P6#=#11111100B           ;バンク1設定
    P6#=#00000001B
    break
case 2:
    P6#=#11111100B           ;バンク2設定
    P6#=#00000010B
    break
case 3:
    P6#=#11111100B           ;バンク3設定
    P6#=#00000011B
ends
RET

;*****
;      LCDイニシャライズ
;*****
LCD_INI:
    !LCD_C=#SYSSET (A)       ;システム・リセット・コマンド
    !LCD_D=#00110000B (A)   ;内部CG使用
    !LCD_D=#87H (A)        ;X方向1文字ドット数

```

```

!LCD_D=#7 (A) ;Y方向1文字ドット数
!LCD_D=#29 (A) ;1行文字数30文字
!LCD_D=#48 (A) ;TC/R=48
!LCD_D=#64 (A) ;1画面ライン数64
!LCD_D=#30 (A) ;表示メモリの1行文字数30文字
!LCD_D=#0 (A)
!LCD_C=#SCROLL (A) ;スクロール・ブロックの設定
!LCD_D=#00 (A) ;第1ブロック・メモリ0000Hから
!LCD_D=#00 (A)
!LCD_D=#64 (A) ;第2ブロック・メモリ1000Hから
!LCD_D=#00 (A)
!LCD_D=#10H (A)
!LCD_D=#64 (A) ;第1, 2画面を反転重ね合わせモードに設定
!LCD_C=#OVLAY (A)
!LCD_D=#0000001B (A)
!LCD_C=#DISPOFF (A) ;画面表示オフの設定
!LCD_D=#0000000B (A)
!LCD_C=#CSRFORM (A) ;カーソル表示の設定
!LCD_D=#0000100B (A) ;ドット数=5
!LCD_D=#0000111B (A) ;文字原点より8ドット目のバー表示
!LCD_C=#HDOTSCR (A)
!LCD_D=#0000000B (A)
CALL !LCDCLR
!LCD_C=#SLEE (A) ;スリープ・モードの設定
RET

;*****
; LCDクリア
;*****
LCDCLR:
!LCD_C=#CSRDIR (A) ;カーソル・アドレス右方向
!LCD_C=#CSRW (A) ;カーソル・アドレスの設定
!LCD_D=#00H (A) ;0000Hに設定
!LCD_D=#00H (A) ;第1ブロックのメモリをクリア
!LCD_C=#WRITE (A)
for(i=#0;i<#240;i++) ;スペース・データの書き込み
!LCD_D=#SPC (A)
next
!LCD_C=#CSRW (A) ;カーソル・アドレスの設定
!LCD_D=#00H (A) ;1000Hに設定
!LCD_D=#10H (A) ;第2ブロックのメモリをクリア
!LCD_C=#WRITE (A)
for(BC=#0;BC<#1920;BC++) (AX) ;全ドット・クリア・データの書き込み
!LCD_D=#00H (A)
next
RET

;*****
; RAMクリア
;*****
RAMCLR:
[HL]=A
HL++ ;アドレス・デクリメント
DBNZ C,$RAMCLR ;設定データ数デクリメント
DBNZ B,$RAMCLR
RET

;*****
; キー・デコード処理
;*****
KEY_DEC:
if_bit(KY_CHGF) ;キー変化あり?
CLR! KY_CHGF ;キー変化フラグ・クリア
SFT_F=SFTKEY (CY) ;シフト・キーの状態を設定
CLR! SFTKEY

HL=#KEYDAT ;キー格納先頭アドレス
CLR! KY_ONF ;キー・コード設定レジスタ
E=#0
for(i=#0;i<#8;i++)
A=#0
if(A!=[HL]) ;キー入力あり?
A=[HL]
for(j=#0;j<#8;j++) ;キー・デコード開始
A>>=1

```

```

        if_bit(CY)                ;テスト対応のキー・オン?
        if_bit(KY_ONF)           ;多重キー?
            SETI    KY_ERRF
        else
            SETI    KY_ONF
            A<->E
            C=A        ;有効データの取り込み
            A<->E
        endif
    endif
    E++
next
else
    A=E                ;キー入力なし
    A+=#8
    E=A
endif
HL++
next
if_bit(KY_ONF)        ;キー入力あり
if_bit(!KY_ERRF)     ;入力キーにエラーなし
    if(KEY_CD==#NOKEY) ;キー・オフ状態あり
        if_bit(!SFT_F) ;シフト・キー・オフ
            HL=#KY_DEC1 ;シフト・オフ状態でキー・デコード
        else
            HL=#KY_DEC2 ;シフト・オン状態でキー・デコード
        endif
        KEY_CD=[HL+C] (A)
        SETI    KY_SIGF
        SETI    LCDCHGF
    endif
endif
else
    KEY_CD=#NOKEY (A) ;キー入力なしの状態を設定
    CLR1    KY_ERRF ;エラー状態の解除
endif
endif
RET

```

```

;*****
; キー・デコード・テーブル
;*****

```

```

KY_DEC1:
DB    'G','N','U',SPC
DB    '0','1','2','3'
DB    'F','M','T',POINT
DB    '4','5','6','7'
DB    'E','L','S','Z'
DB    '8','9',ADD_SIG,SUB_SIG
DB    'D','K','R','Y'
DB    MUL_SIG, DIV_SIG, EQU_SIG, SIG_INV
DB    'C','J','Q','X'
DB    M_ADD, M_SUB, MC, MR
DB    'B','I','P','W'
DB    CUR_LE, CUR_RI, CUR_UP, CUR_DN
DB    'A','H','O','V'
DB    UP, DOWN, MENU, SPC
DB    OFF, DELETE, CLEAR, ENTER
DB    PAR_LE, PAR_RI, COMM, SPC

```

```

KY_DEC2:
DB    'g','n','u',SPC
DB    '0','1','2','3'
DB    'f','m','t',POINT
DB    '4','5','6','7'
DB    'e','l','s','z'
DB    '8','9',ADD_SIG,SUB_SIG
DB    'd','k','r','y'
DB    MUL_SIG, DIV_SIG, EQU_SIG, SIG_INV
DB    'c','j','q','x'
DB    M_ADD, M_SUB, MC, MR
DB    'b','i','p','w'
DB    S_CURL, S_CURR, S_CURU, S_CURD
DB    'a','h','o','v'
DB    S_UP, S_DOWN, MENU, SPC

```

```

DB      OFF_S_DEL, CLEAR, ENTER
DB      PAR_LE, PAR_RI, COMM, SPC

;*****
;          L C D 表示処理
;*****
LCD:
    if_bit(LCDCHGF)
        CLR1    LCDCHGF

        !LCD_C=#CSRW (A)                ;キャラクタ表示エリアの先頭アドレス設定
        !LCD_D=#00H (A)
        !LCD_D=#00H (A)
        !LCD_C=#CSRDIR (A)
        !LCD_C=#MWRITE (A)
        HL=#LCDDS                        ;表示用RAMの先頭アドレスを設定
        for(i=#0;i<#240;i++)           ;表示データ240バイトをLCDに転送
            A=[HL]
            !LCD_D=A
            HL++
        next
        !LCD_C=#CSRW (A)                ;カーソル・アドレスの設定
        !LCD_D=CUR_ADR (A)              ;カーソル・アドレス(下位)の設定
        !LCD_D=#00H (A)                 ;カーソル・アドレス(上位)の設定

        !LCD_C=#DISPON (A)             ;表示オンの設定
        if_bit(CR_DSPF)                 ;カーソルを表示
            !LCD_D=#00010110B (A)       ;カーソル表示オン
        else
            !LCD_D=#00010100B (A)       ;カーソル表示オフ
        endif
    endif

IF_BIT(IV_CHGF)                        ;反転表示行変更
    CLR1    IV_CHGF
    !LCD_C=#CSRW (A)                    ;カーソル・アドレスの設定
    !LCD_D=#00H (A)
    !LCD_D=#10H (A)                     ;1000Hに設定
    !LCD_C=#MWRITE (A)                  ;第2ブロックのメモリをクリア
    for(BC=#0;BC<#1920;BC++) (AX)      ;全ドット・クリア・データの書き込み
        !LCD_D=#00H (A)
    next
    switch(INV_DS)
    case 1:
        !LCD_C=#CSRW (A)                ;1行目を反転表示
        !LCD_D=#00H (A)                 ;開始アドレス1000H番地
        !LCD_D=#10H (A)
        CALL    !INVERT
        break
    case 2:
        !LCD_C=#CSRW (A)                ;2行目を反転表示
        !LCD_D=#0F0H (A)                 ;開始アドレス10F0H番地
        !LCD_D=#10H (A)
        CALL    !INVERT
        break
    case 3:
        !LCD_C=#CSRW (A)                ;3行目を反転表示
        !LCD_D=#0E0H (A)                 ;開始アドレス11E0H番地
        !LCD_D=#11H (A)
        CALL    !INVERT
        break
    case 4:
        !LCD_C=#CSRW (A)                ;4行目を反転表示
        !LCD_D=#0D0H (A)                 ;開始アドレス12D0H番地
        !LCD_D=#12H (A)
        CALL    !INVERT
        break
    case 5:
        !LCD_C=#CSRW (A)                ;5行目を反転表示
        !LCD_D=#0C0H (A)                 ;開始アドレス13C0H番地
        !LCD_D=#13H (A)
        CALL    !INVERT
        break
    case 6:
        !LCD_C=#CSRW (A)                ;6行目を反転表示

```

```

!LCD_D=#0B0H (A)           ;開始アドレス14B0H番地
!LCD_D=#14H (A)
CALL !INVERT
break
case 7:
!LCD_C=#CSRW (A)           ;7行目を反転表示
!LCD_D=#0A0H (A)           ;開始アドレス15A0H番地
!LCD_D=#15H (A)
CALL !INVERT
break
case 8:
!LCD_C=#CSRW (A)           ;8行目を反転表示
!LCD_D=#90H (A)            ;開始アドレス1690H番地
!LCD_D=#16H (A)
CALL !INVERT
ends

!LCD_C=#CSRW (A)           ;カーソル・アドレスの設定
!LCD_D=CUR_ADR (A)         ;カーソル・アドレス(下位)の設定
!LCD_D=#00H (A)           ;カーソル・アドレス(上位)の設定

!LCD_C=#DISPON (A)         ;表示オンの設定
if_bit(CR_DSPF)            ;カーソルを表示
!LCD_D=#00010110B (A)     ;カーソル表示オン
else
!LCD_D=#00010100B (A)     ;カーソル表示オフ
endif
endif

RET
INVERT:
!LCD_C=#CSRDIR (A)
!LCD_C=#WRITE (A)
A=#0FFH
for(i=#0;i<#240;i++)
!LCD_D=A
next
RET

;*****
; 時計カウント処理
;*****
TIME:
IF_BIT(WTIF)               ;0.5秒テスト
CLR! WTIF
IF_BIT(WTISF)              ;1秒テスト
CLR! WTISF

if(MODE0==#5)               ;世界時計モード
SET! LCDCHGF                ;時計表示更新
endif

if(MODE0!=#8)               ;パワーオン状態
AX=POF_CTP                  ;オートパワーオフの時間カウント
AX++
POF_CTP=AX
if(AX)=#AT_OFCT             ;キー変化なし5分経過
MODE0=#8                    ;パワーオフ・モードに設定
if_bit(COMM_F)
CLR! COMM_F                 ;通信状態解除
SET! RS_STN                 ;μPD4711スタンバイに設定
SET! CSINK0                  ;通信使用割り込み禁止
SET! CSINK1
SET! PNK0
endif
endif
endif

SECD++                       ;秒下位インクリメント
if(SECD=#10)
SECD=#0
(SECD+1)++                   ;秒上位インクリメント
if(SECD+1=#8)                ;桁上げあり
(SECD+1)=#0

```



```

MIND++                ;分下位インクリメント
if(MIND==#10)         ;桁上げあり
  MIND=#0
  (MIND+1)++         ;分上位インクリメント
  if(MIND+1==#6)     ;桁上げあり
    (MIND+1)=#0
    HOUD++
    if(HOUD+1!=#02H) ;時データ 24?
      if(HOUD==#10) ;桁上げあり
        HOUD=#0
        (HOUD+1)++
      endif
    elseif(HOUD==#4H) ;時データ 24?
      HOUD=#00H ;日付変更
      HOUD+1=#00H
      DAYD++ ;日データ・インクリメント
      A=YEAD ;現在の月の最終日を参照
      A&=#00000011B
      if_bit(Z) ;うるう年(下位2ビット=0)?
        HL=#CAL_LEA ;うるう年データの参照
      else
        HL=#CAL_NOR ;平年データの参照
      endif
      B=MOND(A)
      B--
      A=[HL+B]
      if(A<DAYD) ;最終日オーバ
        DAYD=#1 ;現在日付を1日に設定
        MOND++ ;月データ・インクリメント
        if(MOND>#12) ;年変更(12月オーバ)
          MOND=#1 ;現在月を1月に設定
          YEAD++ ;年データをインクリメント
          if(YEAD>#199) ;最終年(2099年)オーバ
            YEAD=#1 ;年データの初期化(1901年)
          endif
        endif
      endif
    endif
  endif
endif
endif
endif
endif
endif
ELSE
  SET1  WT1SF ;1秒フラグ・セット
ENDIF
ENDIF

CLR1  ALM_ONF
M_BANK=#0 ;バンク設定
CALL  !BANKSET ;外部メモリ・バンク0設定
CALL  !ALM_SET

M_BANK=#1 ;バンク設定
CALL  !BANKSET ;外部メモリ・バンク1設定
CALL  !ALM_SET

M_BANK=#2 ;バンク設定
CALL  !BANKSET ;外部メモリ・バンク2設定
CALL  !ALM_SET

M_BANK=#3 ;バンク設定
CALL  !BANKSET ;外部メモリ・バンク3設定
CALL  !ALM_SET

if_bit(ALM_ONF) ;アラーム出力?
  if_bit(!SECD.0)
    CLR1  P3.6 ;出力許可
  else
    SET1  P3.6
  endif

  if_bit(STNBYP)
    CLR1  STNBYP
  endif
endif
else

```

```

        SETI      P3.6                ;出力禁止
    endif
    RET

ALM_SET:
    H=#ADRSTA
    L=#00H
    while([HL]<=#SCA_DAT) (A)        ;データ空き領域でない
        if([HL]==#SCA_DAT) (A)      ;アラーム・データ
            if([HL+8]==#ALMON) (A)
                if([HL+1]==#YEAD) (A)
                    if([HL+2]==#MOND) (A)
                        if([HL+3]==#DAYD) (A)

                            AX=HOUDP
                            CALL !ASC_DEL
                            if(A=[HL+4])          ;現在時刻(分) = アラーム設定時間
                                AX=MINDP
                                CALL !ASC_DEL
                                if(A=[HL+5])      ;現在時刻(時) != アラーム設定時間
                                    SETI  ALM_ONF   ;アラーム設定時間あり
                                endif
                            endif
                        endif
                    endif
                endif
            endif
        endif
    endwhile
    CALL !ALM_INC
    if_bit(!CY)
        break
    endif
endw
RET

ALM_INC:
    if(H!=#ADREND) (A)
        H++
        L=#0
        SETI  CY
    else
        CLR1  CY
    endif
    RET

;*****
;          キー入力処理
;*****
INTTM1:
    CLR1  ISP                ;高優先割り込み許可
    EI
    SEL  RB2
    CLR1  KY_STAF
    P3&=#11111000B
    HL=#KEYWK                ;キー・ワーク・エリアのアドレス設定
    for(k=#0;k<#8;k++)
        A=P1
        A^=#11111111B        ;データ反転
        if(A!=[HL])          ;キー変化あり?
            SETI  KY_STAF
            [HL]=A
        endif
        HL++
        A=P3                ;スキャン番号の変更
        A++
        DI                  ;ポート設定中割り込み禁止
        A&=#00000111B
        P3&=#11111000B
        A^=P3
        P3=A
        EI                  ;ポート設定終了割り込み許可
    next

    if_bit(!KY_STAF)          ;キー変化
        if_bit(!KY_ENDF)     ;チャタリング吸収中

```

```

KEYCHT--                                ;チャタリング終了
if(KEYCHT==#0)
  SETI  KY_ENDF
  DE=#KEYWK
  HL=#KEYDAT
  for(k=#0;k<#8;k++)
    if([DE]!=[HL]) (A)                  ;キー変化あり
      SETI  KY_CHGF
      POF_CTP=#0                        ;オートパワーオフの時間カウント・クリア
    endif
    A<->[HL]                            ;KEYWKをKEYDATに転送
    HL++
    DE++
  next
endif
endif
else
  KEYCHT=#3
  CLRI  KY_ENDF
endif
RETI

;*****
;      メニュー処理
;*****
S_MENU:
  if_bit(MD_CGOF)                       ;モード変更あり
    CLRI  MD_CGOF
    SETI  KY_CHGF                        ;キー初期状態設定
    !LCD_C=#SYSSET (A)                  ;システム・リセット・コマンド
    !LCD_D=#00110000B (A)               ;内部CG使用
    HL=#LCDDS                            ;メニュー表示データの設定
    DE=#MENU_DS
    for(i=#0;i<#240;i++)                ;LCD表示エリアにデータ転送
      [HL]=[DE] (A)
      HL++
      DE++
    next
    MODE1=#0                             ;モード初期設定
    MODE2=#0
    MODE3=#0
    MODE4=#0
    SETI  MD_CG1F
    SETI  MD_CG2F
    SETI  MD_CG3F
    SETI  MD_CG4F
    SETI  IV_CHGF                        ;反転表示行を2行目に設定
    INV_DS=#2
    SETI  LCDCHGF
    CLRI  CR_DSPF
  endif
  if_bit(COMM_F)                         ;通信状態
    CALL  !S_COMM                        ;通信処理
    if_bit(!COMM_F)                     ;通信終了
      SETI  MD_CGOF                    ;通信前の状態に設定
    endif
else
  IF_BIT(KY_SIGF)
    CLRI  KY_SIGF
    switch(KEY_CD)                      ;キー・コード分岐
    case '1':                            ;名刺(個人)モードに設定
      MODE0=#1
      SETI  MD_CGOF
      break
    case '2':                            ;名刺(会社)モードに設定
      MODE0=#2
      SETI  MD_CGOF
      break
    case '3':                            ;カレンダー・モードに設定
      MODE0=#3
      SETI  MD_CGOF
      break
    case '4':                            ;メモ帳モードに設定
      MODE0=#4
      SETI  MD_CGOF

```

```

        break
case '5': ;世界時計モードに設定
    MODE0=#5
    SET1 MD_CG0F
    break
case '6': ;初期設定モードに設定
    MODE0=#6
    SET1 MD_CG0F
    break
;
case '7': ;電卓モードに設定
;
;
;
case CUR_UP: ;反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case CUR_DN: ;反転表示行を下に移動
    if(INV_DS<#7)
        INV_DS++
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case ENTER: ;反転表示行のモード実行
    A=INV_DS
    A--
    MODE0=A
    SET1 MD_CG0F ;モード変更状態に設定
    break
case COMM: ;通信状態に設定
    A=ADM_VOL
    A+=ADC_VOL
    A+=MEM_VOL
    COM_VOL=A ;送信データ数の設定
    M_BANK=#0
    CALL !BANKSET
    SET1 TR_CG1F
    SET1 COMM_F

    M_BANK=#0 ;送信メモリのバンク選択
    CALL !BANKSET ;バンク設定
    CALL !MN_COM ;外部メモリ・バンク0設定

    if_bit(ICY)
        M_BANK=#1 ;バンク設定
        CALL !BANKSET ;外部メモリ・バンク1設定
        CALL !MN_COM

        if_bit(ICY)
            M_BANK=#2 ;バンク設定
            CALL !BANKSET ;外部メモリ・バンク2設定
            CALL !MN_COM

            if_bit(ICY)
                M_BANK=#3 ;バンク設定
                CALL !BANKSET ;外部メモリ・バンク3設定
                CALL !MN_COM
                if_bit(ADENDF)
                    TRN_ADR=H (A) ;送信開始アドレスの設定
                    TRN_BNK=M_BANK (A)
                endif
            else
                TRN_ADR=H (A) ;送信開始アドレスの設定
                TRN_BNK=M_BANK (A)
            endif
        else
            TRN_ADR=H (A) ;送信開始アドレスの設定
            TRN_BNK=M_BANK (A)
        endif
    endif
ends
endif
endif

```

```

RET

MN_COM:
HL=#8000H
while([HL]!=#VAC_DAT) (A)           ;データ空き領域でない
  if([HL]<=#MEM_DAT) (A)           ;通信対応データの検索
    SETI  CY
    break
  else
    if(HI=#ADREND) (A)             ;バンク終了
      H++
    else
      M_BANK++
      if(M_BANK>#3)
        SETI  ADENDF
        break
      endif
      CLR1  CY
      break
    endif
  endif
endw
RET

MENU_DS:
DB  'M','E','N','U',SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'1',SPC,'A','D','R','E','S','S'
DB  PAR_LE,'M','A','N',PAR_RI,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'2',SPC,'A','D','R','E','S','S'
DB  PAR_LE,'C','O','M','P','A','N','Y',PAR_RI,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'3',SPC,'C','A','L','E','N','D','A'
DB  'R',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'4',SPC,'M','E','M','O',SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'5',SPC,'C','L','O','C','K',SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,'6',SPC,'S','E','T',SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB  SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

;*****
; データ先頭アドレス検索処理
;*****
STA_SEA:
H=#ADRSTA
L=#00H
M_BANK=#0           ;バンク設定
CALL !BANKSET       ;外部メモリ・バンク0設定
CALL !ADR_SEA
if_bit(ICY)
  M_BANK=#1         ;バンク設定
  CALL !BANKSET     ;外部メモリ・バンク1設定
  CALL !ADR_SEA
if_bit(ICY)
  M_BANK=#2         ;バンク設定
  CALL !BANKSET     ;外部メモリ・バンク2設定
  CALL !ADR_SEA
if_bit(ICY)

```

```

        M_BANK=#3           ;バンク設定
        CALL !BANKSET       ;外部メモリ・バンク3設定
        CALL !ADR_SEA
    endif
endif
endif
RET

;*****
; データ登録アドレス検索処理
;*****
SAVADR:
    CLR1 ADSEAF
    while_bit(!ADSEAF)     ;登録メモリ・アドレス決定
        CALL !ADR_SEA     ;アドレス検索
        SET1 CMPERF
        if_bit(CY)        ;メモリ・バンク変更なし
            if([HL]==B) (A) ;検索データの種類一致
                CALL !DTCOMP ;データ1ライン比較
                if_bit(!CMPERF) ;
                    SET1 ADSEAF ;アドレス決定
                    MENPAG=H (A) ;メモリ・ページの格納
            else
                if_bit(CY)
                    SET1 ADSEAF ;アドレス決定
                    MENPAG=H (A) ;メモリ・ページの格納
                else
                    CALL !ADR_INC
                    if(M_BANK<#4) ;バンク3終了
                        CALL !BANKSET ;バンク変更
                    else
                        break
                    endif
                endif
            endif
        else
            SET1 ADSEAF ;アドレス決定
            MENPAG=H (A) ;メモリ・ページの格納
        endif
    else
        if(M_BANK<#4) ;バンク3終了
            CALL !BANKSET ;バンク変更
        else
            break
        endif
    endif
endw
RET

;*****
; データ・アドレス検索 (前進)
;*****
ADR_SEA:
    CLR1 ADENDF
    while_bit(!ADENDF)
        if([HL]<=B) (A) ;データの検索範囲
            if([HL]==B) (A) ;検索データ一致
                SET1 CY
                SET1 ADENDF
            else
                CALL !ADR_INC
                if_bit(!CY)
                    break
                endif
            endif
        else
            SET1 CY ;データ検索終了
            SET1 ADENDF
        endif
    endw
RET

;*****
; データ・アドレス検索 (後退)
;*****

```

```

AD_SEAR:
  CLR1  ADENDF
  while_bit(!ADENDF)
    if([HL]<B) (A) ;データの検索範囲
      if([HL]==B) (A) ;検索データ一致
        SET1  CY
        SET1  ADENDF
      else
        CALL  !ADR_DEC
        if_bit(CY)
          break
        endif
      endif
    else
      SET1  CY ;検索データ終了
      SET1  ADENDF
    endif
  endw
  RET

;*****
; データ1ライン比較
;*****
DTCOMP:
  CLR1  CMPERF
  PUSH  BC
  PUSH  DE
  for(i=#0;i<#30;i++) ;1ライン(30文字)の比較データ数を設定
    A=[DE]
    if(A==[HL+C]) ;登録データと外部メモリデータの比較
      DE++ ;データ一致
      C++
    else ;データ不一致処理
      SET1  CMPERF
      break ;外部メモリが大きいときCY=1
    endif
  next
  POP   DE
  POP   BC
  RET

;*****
; データ登録
;*****
DT_SAV:
  while(M_BANK<#4) ;データ登録終了
    CALL  !DTCHNG
    if(H>#ADREND) (A) ;バンク最終アドレス
      H=#ADRSTA ;開始アドレスの設定
      L=#0
      E=#0
      M_BANK++
      if(M_BANK<#4) ;バンク3終了
        CALL  !BANKSET ;バンク変更
      endif
    endif
  endw
  RET

;*****
; データ交換
;*****
DTCHNG:
  for(i=#0;i<#255;i++) ;1ページ・データ転送
    A=[HL]
    A<->[DE]
    [HL]=A
    HL++ ;ポインタ変更
    E++
  next
  A=[HL]
  A<->[DE]
  [HL]=A
  HL++

```

```

E++
RET

;*****
;      データ削除
;*****
DT_DEL:
D=H (A)
L=#0
D++
E=#0
WHILE(M_BANK<#4)
IF(HI=#ADREND) (A)
for(i=#0;i<#255;i++)
[HL]=[DE] (A)
HL++
DE++
next
[HL]=[DE] (A)
HL++
DE++
ELSE
M_BANK++
if(M_BANK<#4)
CALL !BANKSET
D=#ADRSTA
E=#0
HL=#DTWORK
for(i=#0;i<#255;i++)
[HL]=[DE] (A)
HL++
DE++
next
[HL]=[DE] (A)
M_BANK--
CALL !BANKSET
H=#ADREND
L=#0
DE=#DTWORK
for (i=#0;i<#255;i++)
[HL]=[DE] (A)
HL++
DE++
next
[HL]=[DE] (A)
M_BANK++
CALL !BANKSET
H=#ADRSTA
L=#0
D=#ADRSTA+1
E=#0
else
A=#VAC_DAT
for(i=#0;i<#255;i++)
[HL]=A
HL++
next
[HL]=A
endif
ENDIF
ENDW
RET

;*****
;      1ライン・データ削除(休日データ用)
;*****
DEL_IL:
while(forever)
if(LI=#OF0H) (A)
[HL]=[HL+10H] (A)
L++
ELSEIF(H==#ADREND) (A)
if(M_BANK<#3)
M_BANK++
CALL !BANKSET

```

```

;データ登録終了
;バンク最終アドレス
;1ページ・データ転送

```

```

;バンク変更
;バンク3終了
;バンク先頭アドレスをワーク・エリアに転送

```

```

;前バンクの最終アドレスを設定
;ワーク・エリアのデータを最終アドレスに転送

```

```

;バンクを次バンクに設定
;転送先の初期アドレスに設定
;転送元の初期アドレスに設定
;バンク3最終アドレスを空き状態に設定

```

```

;1頁内の最終行
;次の行データ(+10H)参照
;バンク内の最終アドレス
;最終バンクでない
;次のバンクに設定

```



```

D=#ADRSTA ;先頭アドレスの設定
E=#0
if([DE]==#HOL_DAT) (A) ;先頭データが休日データ
HL=#DTWORK ;ワーク・エリアの設定
for(i=#0;i<#10H;i++) ;1行分のデータをワークに転送
[HL]=[DE] (A)
HL++
DE++
next
M_BANK-- ;前のバンクに設定
CALL !BANKSET
D=#ADREND ;最終アドレスに設定
E=#0FOR
for(i=#0;i<#10H;i++) ;最終行にデータ登録
[DE]=[HL] (A)
DE++
HL++
next
H=#ADRSTA ;次バンクの先頭ページに設定
L=#0
M_BANK++
CALL !BANKSET
if([HL+10H]==#VAC_DAT) (A) ;休日データなし
CALL !DT_DEL ;データ削除
break
endif
else
M_BANK-- ;前のメモリ・バンクに設定
CALL !BANKSET
H=#ADREND ;バンク内の最終アドレスに設定
L=#0FOR
A=#VAC_DAT ;データなし状態に設定
for(i=#0;i<#10H;i++) ;最終行をデータなしに設定
[HL]=A
HL++
next
break
endif
else
A=#VAC_DAT ;データなし状態に設定
for(i=#0;i<#10H;i++) ;最終行をデータなしに設定
[HL]=A
HL++
next
break
endif
else
if([HL+10H]!=#HOL_DAT) (A) ;次ページはタイトル・データでない
A=#VAC_DAT
for(i=#0;i<#10H;i++) ;最終行を空きデータに設定
[HL]=A
HL++
next
break
else
for(i=#0;i<#10H;i++) ;次ページ先頭データを最終行に設定
[HL]=[HL+10H] (A)
HL++
next
if([HL+10H]==#VAC_DAT) (A) ;休日データなし
CALL !DT_DEL ;データ削除
break
endif
endif
endif
endw
RET

;*****
; データ検索 (アドレス前進)
;*****
SEA_FOR:
CLR! ADSEAF
while_bit(!ADSEAF) ;データ検索
CALL !ADR_SEA

```

```

if_bit(CY)                                ;バンク変更
  if([HL]==B) (A)                          ;検索データの種類一致
    if([HL+C]==SEACHR) (A)                 ;検索文字データ一致
      SETI CY
      SETI ADSEAF
      MEMPAG=H (A)                         ;検索アドレスをメモリに格納
      MEMBNK=M_BANK (A)
    else
      CALL IADR_INC
      if_bit(!CY)
        if(M_BANK<=#3)
          CALL !BANKSET                    ;バンク変更
        else
          break
        endif
      endif
    endif
  else
    break
  endif
endif
else
  if(M_BANK<=#3)
    CALL !BANKSET                          ;バンク変更
  else
    break
  endif
endif
endw
RET

;*****
; データ検索 (アドレス後退)
;*****
SEA_REV:
  CLRI ADSEAF
  while_bit(!ADSEAF)                       ;データ検索
    CALL !AD_SEAR
    if_bit(CY)                              ;バンク変更
      if([HL]==B) (A)                       ;検索データの種類一致
        if([HL+C]==SEACHR) (A)             ;検索文字データ一致
          SETI CY
          SETI ADSEAF
          MEMPAG=H (A)                     ;検索アドレスをメモリに格納
          MEMBNK=M_BANK (A)
        else
          CALL IADR_DEC
          if_bit(!CY)
            if(M_BANK<=#3)
              CALL !BANKSET                ;バンク変更
            else
              break
            endif
          endif
        endif
      else
        break
      endif
    else
      if(M_BANK<=#3)
        CALL !BANKSET                      ;バンク変更
      else
        break
      endif
    endif
  endw
  RET

;*****
; 1行データ検索 (アドレス前進)
;*****
SEA_FIL:

```

```

CLR1 ADSEAF
while_bit(!ADSEAF) ;データ検索
CALL !ADR_SEA
if_bit(CY) ;バンク変更
if([HL]=B) (A) ;検索データの種類一致
CALL !DTCOMP ;検索文字データ一致
if_bit(!CMPERF)
SET1 ADSEAF
MEMPAG=H (A) ;検索アドレスをメモリに格納
MEMBNK=M_BANK (A)
else

CALL !ADR_INC
if_bit(!CY)
if(M_BANK<=#3)
CALL !BANKSET ;バンク変更
else
break
endif
endif

endif
else
break
endif
else
if(M_BANK<=#3)
CALL !BANKSET ;バンク変更
else
break
endif
endif
endif
endw
RET

;*****
; 1行データ検索 (アドレス後退)
;*****
SEA_R1L:
CLR1 ADSEAF
while_bit(!ADSEAF) ;データ検索
CALL !AD_SEAR
if_bit(CY) ;バンク変更
if([HL]=B) (A) ;検索データの種類一致
CALL !DTCOMP ;検索文字データ一致
if_bit(!CMPERF)
SET1 ADSEAF
MEMPAG=H (A) ;検索アドレスをメモリに格納
MEMBNK=M_BANK (A)
else

CALL !ADR_DEC
if_bit(!CY)
if(M_BANK<=#3)
CALL !BANKSET ;バンク変更
else
break
endif
endif

endif
else
break
endif
else
if(M_BANK!=#0)
CALL !BANKSET ;バンク変更
else
break
endif
endif
endif

```

```

    endw
    RET

;*****
;   アドレス・インクリメント処理
;*****
ADR_INC:
    if(H1=#ADREND) (A)           ;バンク終了
        H++                       ;アドレス・インクリメント
        SET1  CY
    else
        H=#ADRSTA                 ;スタート・アドレスを設定
        M_BANK++
        CLR1  CY
    endif
    RET

;*****
;   アドレス・デクリメント処理
;*****
ADR_DEC:
    if(H1=#ADRSTA) (A)          ;バンク終了
        H--                       ;アドレス・デクリメント
        SET1  CY
    else
        H=#ADREND                ;スタート・アドレスを設定
        M_BANK--
        CLR1  CY
    endif
    RET

;*****
;   次の検索アドレス設定処理
;*****
NEX_ADR:
    if(M_BANK==#0)
        if(H1=#ADRSTA) (A)
            CALL  !ADR_DEC
            REV_ADR=H (A)
            REV_BNK=M_BANK (A)
            CALL  !ADR_INC
        endif
    else
        CALL  !ADR_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
        CALL  !ADR_INC
    endif
    RET

;*****
;   前の検索アドレス設定処理
;*****
BEF_ADR:
    if(M_BANK==#3)
        if(H1=#ADREND) (A)
            CALL  !ADR_INC
            FOR_ADR=H (A)
            FOR_BNK=M_BANK (A)
            CALL  !ADR_DEC
        endif
    else
        CALL  !ADR_INC
        FOR_ADR=H (A)
        FOR_BNK=M_BANK (A)
        CALL  !ADR_DEC
    endif
    RET

;*****
;   パワーオフ処理
;*****
P_OFF:
    DI                               ;割り込み禁止
    SET1  THNK1                       ;8ビット・タイマ1 割り込み禁止

```

```

CLR1  PMK3                ;INTP3 (ONキー) の割り込み許可
CLR1  WTMK                ;時計用タイマの割り込み許可
CLR1  PIF3                ;INTP3 (ONキー) の割り込み要求フラグ・クリア
SET1  STNBYF              ;スタンバイ中の状態に設定
SET1  RS_STN              ;μPD4711をスタンバイ・モードに設定
!LCD_C=#DISPOFF (A)      ;画面表示オフの設定
!LCD_D=#00000000B (A)
CALL  !LCDCLR             ;LCD表示クリア
LCD_C=#SLEE (A)          ;LCDスリープ状態に設定
repeat
  STOP                    ;STOPモードに設定
  if_bit(WTIF)            ;時計用タイマ・オーバーフロー発生
  CALL  !TIME              ;時計カウント
  if_bit(!STNBYF)        ;アラーム設定時間一致あり
  break
endif
until_bit(PIF3)          ;ONキー入力あり

SET1  PMK3                ;INTP3 (ONキー) の割り込み禁止
SET1  WTMK                ;時計用タイマの割り込み禁止
CLR1  TMMK1               ;8ビット・タイマ1割り込み許可
EI

MODE0=#0
SET1  MD_CGOF
POF_CTP=#0
RET

;*****
; LCD表示エリアのクリア
;*****
LCD_DSC:
  A=#SPC
  HL=#LCDDS
  for(i=#0;i<#240;i++)
    [HL]=A
    HL++
  next
  RET

;*****
; メモリ圏のデータ転送
;*****
DT_SET:
  [HL]=[DE] (A)
  HL++
  DE++
  DBNZ  B, $DT_SET
  RET

;*****
; LCD 1行データ・セット
;*****
DT1LSET:
  for(i=#0;i<#30;i++)
    [HL]=[DE] (A)
    HL++
    DE++
  next
  RET

;*****
; LCD 1ページ・データ・セット
;*****
DT1PSET:
  for(i=#0;i<#240;i++)
    [HL]=[DE] (A)
    HL++
    DE++
  next
  RET

;*****
; 2バイト表示コード→1バイト・データ変換

```

;*****

ASC_DEL:

A&=#0FH	:上位データの低位4ビット有効
A<<=1	:数値データを上位4ビットに設定
A<<=1	
A<<=1	
A<<=1	
A<->X	:低位データの取り込み
A&=#0FH	:低位4ビットを有効
A =X	:1バイト変換データの設定
RET	
END	

\$PC(014)

```
EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
```

```
*****
; 名刺(個人)LCD表示データ
*****
PUBLIC  ADM_D, REG_D, COR_D, SEA_D, DEL_D, CLRILD, NAME_D, TEL_D, FAX_D, ADR_D
PUBLIC  COMP_D, MEMO_D, NA_SEA, ATOM_D, NTOZ_D, NEXT_P, NAN_DAT, MEM_OV

*****
; 名刺(会社)LCD表示データ
*****
PUBLIC  ADC_D, COM_D

*****
; カレンダーLCD表示データ
*****
PUBLIC  CALEND, TOD_D, MONTH_D, SETHOL1, WEK_D, OPEN_D, END_D, ALARM_D, MEMO_D1
PUBLIC  BUSI_D, YEAR_D1, YEAR_D2, MON_D1, MON_D2, DAY_D, SETHOL2, WEKHOL1
PUBLIC  SETDAT1, WEKHOL2, HOLMAK, WEKMAK, SETDAT2, DELKEY

*****
; メモ帳LCD表示データ
*****
PUBLIC  MEMODSP, TIT_L1, REG_L2, TIT_SEA, T_DAT

*****
; 世界時計LCD表示データ
*****
PUBLIC  CLOCK, DISPLAY, CIT_SEL, CITY_D, NEWYORK, LOSANG, BRASIL, LIMA, LONDON
PUBLIC  ROME, CAIRO, MOSCOW, HONOLU, SYDNEY, PEKING, HONG, NEWDEL, KUWAIT

*****
; 初期設定LCD表示データ
*****
PUBLIC  SET_D, DATE_D, TIME_D, SETDATE, YEAR, MONTH, DAY, SETTIME, HOUR, MINUTE, SECOND

*****
; 通信処理表示データ
*****
PUBLIC  COMMUN, TRANS, RECEIVE, PC9801, ELE_POC
```

LCDSEG CSEG

```
*****
; 名刺(個人)LCD表示データ
*****
ADM_D:
DB      'A', 'D', 'D', 'R', 'E', 'S', 'S', PAR_LE, 'N', 'A'
DB      'N', PAR_RI, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

REG_D:
DB      SPC, SPC, 'R', 'E', 'G', 'I', 'S', 'T', 'E', 'R'
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COR_D:
DB      SPC, SPC, 'C', 'O', 'R', 'R', 'E', 'C', 'T', 'I'
DB      'O', 'N', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SEA_D:
DB      SPC, SPC, 'S', 'E', 'A', 'R', 'C', 'R', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DEL_D:
DB      SPC, SPC, 'D', 'E', 'L', 'E', 'T', 'E', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

CLRILD:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
```



```

DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NAME_D:
DB      'N', 'A', 'M', 'E', QUEST, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TEL_D:
DB      'T', 'E', 'L', QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

FAX_D:
DB      'F', 'A', 'X', QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

ADR_D:
DB      'A', 'D', 'D', 'R', 'E', 'S', 'S', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COMP_D:
DB      'C', 'O', 'M', 'P', 'A', 'N', 'Y', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MEMO_D:
DB      'M', 'E', 'M', 'O', QUEST, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NA_SEA:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'N', 'A', 'M', 'E', SPC, 'S', 'E', 'A', 'R', 'C'
DB      'H', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

ATOM_D:
DB      SPC, SPC, 'A', SPC, 'B', SPC, 'C', SPC, 'D', SPC
DB      'E', SPC, 'F', SPC, 'G', SPC, 'H', SPC, 'I', SPC
DB      'J', SPC, 'K', SPC, 'L', SPC, 'M', SPC, SPC, SPC

NTOZ_D:
DB      SPC, SPC, 'N', SPC, 'O', SPC, 'P', SPC, 'Q', SPC
DB      'R', SPC, 'S', SPC, 'T', SPC, 'U', SPC, 'V', SPC
DB      'W', SPC, 'X', SPC, 'Y', SPC, 'Z', SPC, SPC, SPC

NEXT_P:
DB      SPC, SPC, 'N', 'E', 'X', 'T', SPC, 'P', 'A', 'G'
DB      'E', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NAM_DAT:
DB      'N', 'A', 'M', 'E', SPC, 'D', 'A', 'T', 'A', SPC
DB      PAR_LE, SPC, PAR_RI, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MEM_OV:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, 'M'
DB      'E', 'M', 'O', 'R', 'Y', SPC, 'O', 'V', 'E', 'R'
DB      'F', 'L', 'O', 'W', SPC, SPC, SPC, SPC, SPC, SPC

;*****
;   名刺(会社)LCD表示データ
;*****
ADC_D:
DB      'A', 'D', 'D', 'R', 'E', 'S', 'S', PAR_LE, 'C', 'O'
DB      'M', 'P', 'A', 'N', 'Y', PAR_RI, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COM_D:
DB      'C', 'O', 'M', 'P', 'A', 'N', 'Y', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
    
```



```

;*****
; カレンダーLCD表示データ
;*****
CALEND:
DB 'C','A','L','E','N','D','A','R',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

TOD_D:
DB SPC,SPC,'T','O','D','A','Y',SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MONTH_D:
DB SPC,SPC,'M','O','N','T','H',SPC,'D','I'
DB 'S','P','L','A','Y',SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

SETHOL1:
DB SPC,SPC,'S','E','T','H','O','L','I'
DB 'D','A','Y',SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

WEEK_D:
DB SPC,'S','U','N','M','O','N',SPC,'T'
DB 'U','E',SPC,'W','E','D',SPC,'T','H','U'
DB SPC,'F','R','I','S','A','T',SPC,SPC

OPEN_D:
DB SPC,SPC,'O','P','E','N',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

END_D:
DB SPC,SPC,'E','N','D',QUEST,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

ALARM_D:
DB SPC,SPC,'A','L','A','R','M',QUEST,SPC,SPC
DB SPC,SPC,'O','N',SPC,DIV_SIG,SPC,'O','F','F'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

BUSI_D:
DB SPC,SPC,'B','U','S','I','N','E','S','S'
DB QUEST,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MEMO_D1:
DB SPC,SPC,'M','E','M','O',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

YEAR_D1:
DB 'Y','E','A','R',QUEST,PAR_LE,'1','9','0'
DB '1',SUB_SIG,'2','0','9',PAR_RI,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

YEAR_D2:
DB SPC,SPC,'Y','E','A','R',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MON_D1:
DB 'M','O','N','T','H',QUEST,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MON_D2:
DB SPC,SPC,'M','O','N','T','H',QUEST,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DAY_D:

```

```

DB      SPC, SPC, 'D', 'A', 'Y', QUEST, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETHOL2:
DB      'S', 'E', 'T', SPC, 'H', 'O', 'L', 'I', 'D', 'A'
DB      'Y', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

WEKHOL1:
DB      SPC, SPC, 'W', 'E', 'E', 'K', 'L', 'Y', SPC, 'H'
DB      'O', 'L', 'I', 'D', 'A', 'Y', SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETDAT1:
DB      SPC, SPC, 'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E'
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

WEKHOL2:
DB      'W', 'E', 'E', 'K', 'L', 'Y', SPC, 'H', 'O', 'L'
DB      'I', 'D', 'A', 'Y', SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

HOLMAK:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'O', COLON, 'H', 'O', 'L', 'I', 'D', 'A', 'Y', SPC

WEKMAK:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'X', COLON, 'W', 'E', 'E', 'K', 'L', 'Y', SPC

SETDAT2:
DB      'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DELKEY:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, 'D', 'E', 'L', SPC
DB      'K', 'E', 'Y', EQU_SIG, 'C', 'L', 'E', 'A', 'R'

;*****
;      メモ帳LCD表示データ
;*****
MEMODSP:
DB      'M', 'E', 'M', 'O', SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIT_L1:
DB      'T', 'I', 'T', 'L', 'E', PAR_LE, 'L', 'E', 'V', 'E'
DB      'L', 'I', PAR_RI, QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

REG_L2:
DB      'R', 'E', 'G', 'I', 'S', 'T', 'E', 'R', PAR_LE, 'L'
DB      'E', 'V', 'E', 'L', '2', PAR_RI, QUEST, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIT_SEA:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, 'T'
DB      'I', 'T', 'L', 'E', SPC, 'S', 'E', 'A', 'R', 'C'
DB      'H', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

T_DAT:
DB      'T', 'I', 'T', 'L', 'E', SPC, 'D', 'A', 'T', 'A'
DB      PAR_LE, SPC, PAR_RI, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

;*****
;      世界時計LCD表示データ
;*****

```

```

CLOCK:
DB 'C','L','O','C','K',SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DISPLAY:
DB SPC,SPC,'D','I','S','P','L','A','Y',SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CIT_SEL:
DB SPC,SPC,'C','I','T','Y',SPC,'S','E','L'
DB 'E','C','T',SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CITY_D:
DB 'C','I','T','Y',SPC,'S','E','L','E','C'
DB 'T',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

NEWYORK:
DB SPC,SPC,'N','E','W',SPC,'Y','O','R','K'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LOSANG:
DB SPC,SPC,'L','O','S',SPC,'A','N','G','E'
DB 'L','E','S',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

BRASIL:
DB SPC,SPC,'B','R','A','S','I','L','I','A'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LINA:
DB SPC,SPC,'L','I','N','A',SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LONDON:
DB SPC,SPC,'L','O','N','D','O','N',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

ROME:
DB SPC,SPC,'R','O','M','E',SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CAIRO:
DB SPC,SPC,'C','A','I','R','O',SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MOSCOW:
DB SPC,SPC,'M','O','S','C','O','W',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

HONOLU:
DB SPC,SPC,'H','O','N','O','L','U','L','U'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

SYDNEY:
DB SPC,SPC,'S','Y','D','N','E','Y',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

PEKING:
DB SPC,SPC,'P','E','K','I','N','G',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

```



```

HONG:
DB   SPC, SPC, 'H', 'O', 'N', 'G', SPC, 'K', 'O', 'N'
DB   'G', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NEWDEL:
DB   SPC, SPC, 'N', 'E', 'W', SPC, 'D', 'E', 'L', 'H'
DB   'I', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

KUWAIT:
DB   SPC, SPC, 'K', 'U', 'W', 'A', 'I', 'T', SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

;*****
;   初期設定LCD表示データ
;*****
SET_D:
DB   'S', 'E', 'T', SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DATE_D:
DB   SPC, SPC, 'D', 'A', 'T', 'E', SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIME_D:
DB   SPC, SPC, 'T', 'I', 'M', 'E', SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETDATE:
DB   'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E', SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

YEAR:
DB   SPC, SPC, 'Y', 'E', 'A', 'R', QUEST, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MONTH:
DB   SPC, SPC, 'M', 'O', 'N', 'T', 'H', QUEST, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DAY:
DB   SPC, SPC, 'D', 'A', 'Y', QUEST, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETTIME:
DB   'S', 'E', 'T', SPC, 'T', 'I', 'M', 'E', SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

HOUR:
DB   SPC, SPC, 'H', 'O', 'U', 'R', QUEST, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MINUTE:
DB   SPC, SPC, 'M', 'I', 'N', 'U', 'T', 'E', QUEST, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SECOND:
DB   SPC, SPC, 'S', 'E', 'C', 'O', 'N', 'D', QUEST, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

;*****
;   通信処理表示データ

```

;*****

COMMUN:

```
DB 'C','O','M','M','U','N','I','C','A','T'
DB 'I','O','N',SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

TRANS:

```
DB SPC,SPC,'T','R','A','N','S','M','I','T'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

RECEIVE:

```
DB SPC,SPC,'R','E','C','E','I','V','E',SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

PC9801:

```
DB SPC,SPC,'P','C',SUB_SIG,'9','8','0','1',PAR_LE
DB 'R','S',SUB_SIG,'2','3','2','C',PAR_RI,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

ELE_POC:

```
DB SPC,SPC,'E','L','E','C','T','R','O','N'
DB 'I','C',SPC,'P','O','C','E','T','B'
DB 'O','O','K',SPC,SPC,SPC,SPC,SPC,SPC,SPC
END
```

LCDCMD. INC

```

*****
;
;       LCDコマンド定義
*****
SYSSET EQU          0100000B
SCROLL EQU          01000100B
OVLAY EQU           01011011B
CSRFORM EQU         01011101B
CSRDIR EQU          01001100B
CSRW EQU            01000110B
MWRITE EQU          01000010B
SLEE EQU            01010011B
DISPON EQU          01011001B
DISPOFF EQU         01011000B
HDOTSCR EQU         01011010B

```

LCDT_EX. INC

```

*****
;
;       名刺(個人) LCD表示データ
*****
EXTRN  ADM_D, REG_D, COR_D, SEA_D, DEL_D, CLR1LD, NAME_D, TEL_D, FAX_D, ADR_D
EXTRN  COMP_D, MEMO_D, NA_SEA, ATOM_D, NTOZ_D, NEXT_P, NAM_DAT, MEM_OV

;
;       名刺(会社) LCD表示データ
*****
EXTRN  ADC_D, COM_D

;
;       カレンダー LCD表示データ
*****
EXTRN  CALEND, TOD_D, MONTH_D, SETHOL1, WEEK_D, OPEN_D, END_D, ALARM_D, MEMO_D1
EXTRN  BUSI_D, YEAR_D1, YEAR_D2, MON_D1, MON_D2, DAY_D, SETHOL2, WEKHOL1
EXTRN  SETDAT1, WEKHOL2, HOLMAK, WEKMAK, SETDAT2, DELKEY

;
;       メモ帳 LCD表示データ
*****
EXTRN  MEMODSP, TIT_L1, REG_L2, TIT_SEA, T_DAT

;
;       世界時計 LCD表示データ
*****
EXTRN  CLOCK, DISPLAY, CIT_SEL, CITY_D, NEWYORK, LOSANG, BRASIL, LIMA, LONDON
EXTRN  ROME, CAIRO, MOSCOW, HONOLU, SYDNEY, PEKING, HONG, NEWDEL, KUWAIT

;
;       初期設定 LCD表示データ
*****
EXTRN  SET_D, DATE_D, TIME_D, SETDATE, YEAR, MONTH, DAY, SETTIME, HOUR, MINUTE, SECOND

;
;       通信処理表示データ
*****
EXTRN  COMMUN, TRANS, RECEIVE, PC9801, ELB_POC

```

PORT. INC

```

*****
;
;       ポート定義
*****
BUSYIN EQU          P2.3           ;電子手帳通信ビジィ信号入力
BUSYOUT EQU         P2.4          ;電子手帳通信ビジィ信号出力
RS_STN EQU          P3.5          ;μ PD4711スタンバイ信号出力
CTS EQU            P3.3           ;RS-232C CTS入力
RTS EQU            P3.4           ;RS-232C RTS出力
A16 EQU            P6.0           ;外部メモリ・アドレス
A17 EQU            P6.1           ;外部メモリ・アドレス
RS_STA EQU          P0.0          ;RS-232C スタート・ビット検出
PW_ON EQU           P0.3          ;O Nキー検出

```

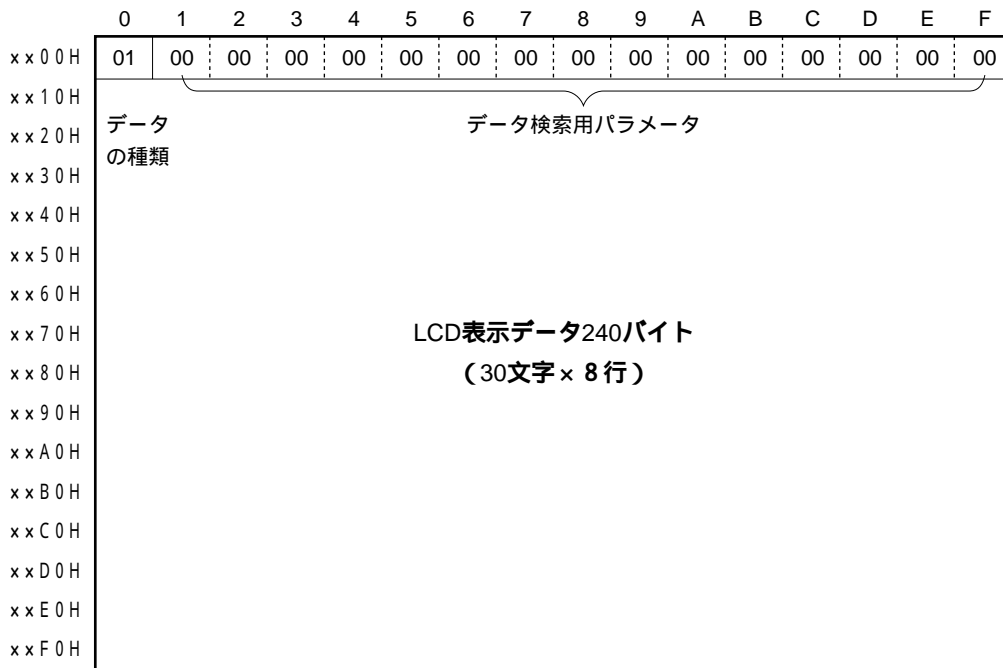
4.4 名刺（個人）モード

名刺（個人）モードでは、モード選択や、データの登録、修正、検索、削除を行っています。

4.4.1 登録データのフォーマット

名刺（個人）データの登録フォーマットを図4 - 3に示します。

図4 - 3 名刺（個人）データの登録フォーマット



4.4.2 モード選択条件

名刺（個人）モードの遷移状態の条件を表4 - 5 に示します。

表4 - 5 名刺（個人）モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
1	0	-	-	-	名刺（個人）モード選択
	1	0	-	-	名刺（個人）名前登録モード
		1	-	-	名刺（個人）電話番号登録モード
		2	-	-	名刺（個人）FAX番号登録モード
		3	-	-	名刺（個人）住所登録モード
		4	-	-	名刺（個人）会社名登録モード
		5	-	-	名刺（個人）メモ内容登録モード
		6	-	-	メモリ・オーバフロー表示モード
	2	0	-	-	先頭文字選択モード
		1	-	-	名前選択モード
		2	-	-	データ修正モード
	3	0	-	-	先頭文字選択モード
		1	-	-	名前選択モード
		2	-	-	データ表示モード
		3	-	-	会社データ表示モード
	4	0	-	-	先頭文字選択モード
		1	-	-	名前選択モード
		2	-	-	データ表示モード

備考 - : don't care

4.4.3 登録処理

登録モードでは、ENTキー入力によりMODE2の値を更新し、CLRキー入力のときは前モードに戻すためMODE1を0に設定します。登録データは、文字データのキー入力により、カーソルのあるアドレス(CUR_ADR)に文字データを書き込みます。すべてのデータ入力後(メモ入力後のENTキー)、次の順で登録処理を行います。

メモリ最終ページの先頭アドレスの確認をします。

バンク3(P61, P60 = 1, 1)のBF00H番地がFFH(空きエリア)以外のときはメモリ・オーバーフローにします(MODE2 = 6に設定)。

データを登録するアドレスを検索します。

検索条件: $\times \times 00H$ が02H以上、または、 $\times \times 00H$ が01Hで入力した名前のデータが、表示コードで比較して小さい値のとき

検索したアドレスにデータを登録します。表示データ(240文字)は、 $\times \times 10H - \times \times FFH$ に登録し、 $\times \times 00H$ に01Hを、 $\times \times 01H - \times \times 0FH$ に00Hに登録します。検索アドレス以降のデータは、次のアドレス(+100H)にシフトします。

データ数(ADM_VOL)を+1します。

4.4.4 データ選択処理

データ選択処理には、先頭文字選択モード(MODE2 = 0)と、名前選択モード(MODE2 = 1)の2種類があります。

(1) 先頭文字選択モード

先頭文字選択モードでは、`←`、`→`、`↑`、`↓` キー入力により、カーソルのアドレス(CUR_ADR)の値に加算、減算をしてカーソルの位置を変更しています。ENTキー入力により、先頭文字を決定し(SEACHRにA-Zを設定)、名前選択モードに設定します(MODE2 = 1)。ただし、NEXT PAGEを選択したときは、データ登録メモリの $\times \times 00H$ に01Hが格納されている先頭アドレスを検索し、次のモード(MODE2 = 2)に移行します。検索データがないときは、名刺(個人)モード選択に戻ります(MODE1 = 0)。

図4-4 名刺(個人)データの検索

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\times \times 00H$	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

01Hで格納されている先頭を検索

(2) 名前選択モード

名前選択モードでは、データ登録メモリの $\times \times 00\text{H} = 01\text{H}$ で、 $\times \times 10\text{H} = \text{SEACHR}$ の名前データ ($\times \times 10\text{H} - \times \times 1\text{FH}$) を検索して表示します。検索データは、1回の処理で7個(表示可能な範囲)まで表示します。表示している名前の先頭/最終アドレスをメモリ (REV_ADR/FOR_ADR) に格納し、次の検索時の開始アドレスとして使用します。

UP/DOWNキーにより、検索文字のデータ (SEACHR) を + 1/ - 1 しています。検索文字変更後は、データ登録メモリの先頭から開始します。

ENTキー入力により、反転表示している名前と $\times \times 10\text{H}$ から30文字のデータが一致しているアドレスを検索し、データ表示のモードに移行します (MODE2 = 2)。

図4-5 名刺(個人)の名前データの検索

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\times \times 00\text{H}$	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
$\times \times 10\text{H}$	× × 名前データ (30文字)															
$\times \times 20\text{H}$	SEACHR															

4.4.5 修正処理

修正モードでは、ENTキー入力により修正完了とし、修正前のデータ登録メモリのアドレスに再登録します。登録後は、名刺(個人)モード選択に戻します (MODE1 = 0)。CLRキー入力のときは、前モードに戻すためMODE1を0に設定します。修正は、文字データのキー入力により、カーソルのあるアドレス (CUR_ADR) に文字データを上書きしています。

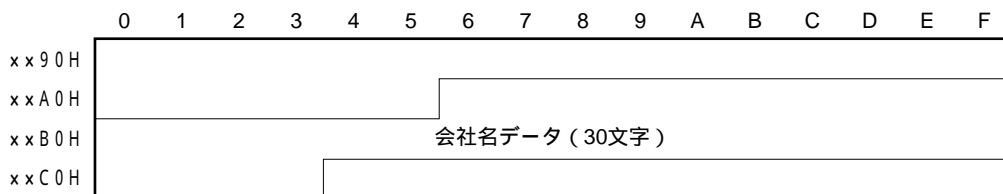
修正データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの + 100H/ - 100H の $\times \times 00\text{H}$ データが01Hのときは、表示データの変更をします。

4.4.6 検索処理

検索モードでは、ENTキー入力により、会社データ表示モードになります。会社データ表示モードでは、名刺（個人）の会社名のデータ（図4-6参照）と、名刺（会社）データ（ $\times \times 00H = 02H$ ）の $\times \times 10H$ から30文字のデータが一致しているアドレスを検索し、会社データ表示のモードに移行します（MODE2 = 3）。

検索データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの+ 100H/- 100Hの $\times \times 00H$ データが01Hのときは、表示データの変更をします。

図4-6 名刺（個人）の会社名データ



4.4.7 削除処理

削除モードでは、DELキー入力により表示中のデータを削除します。データ削除後は、個人（名刺）モード選択に戻します（MODE1 = 0）。CLRキー入力の際は、前モードに戻すためMODE1を0に設定します。

削除データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの+ 100H/- 100Hの $\times \times 00H$ データが01Hのときは、表示データの変更をします。次の順でデータ削除を行っています。

削除データのアドレスを検索します。

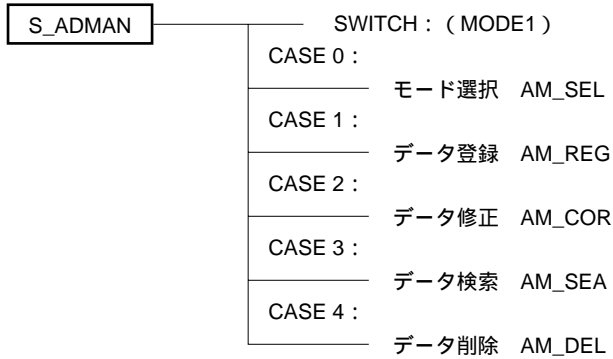
検索したアドレスを、次のアドレス（+ 100H）データから最終アドレス（P61, P60 = 1, 1, BFFFH）までシフトし、上書きすることでデータを削除します。

データ数（ADM_VOL）を - 1 します。

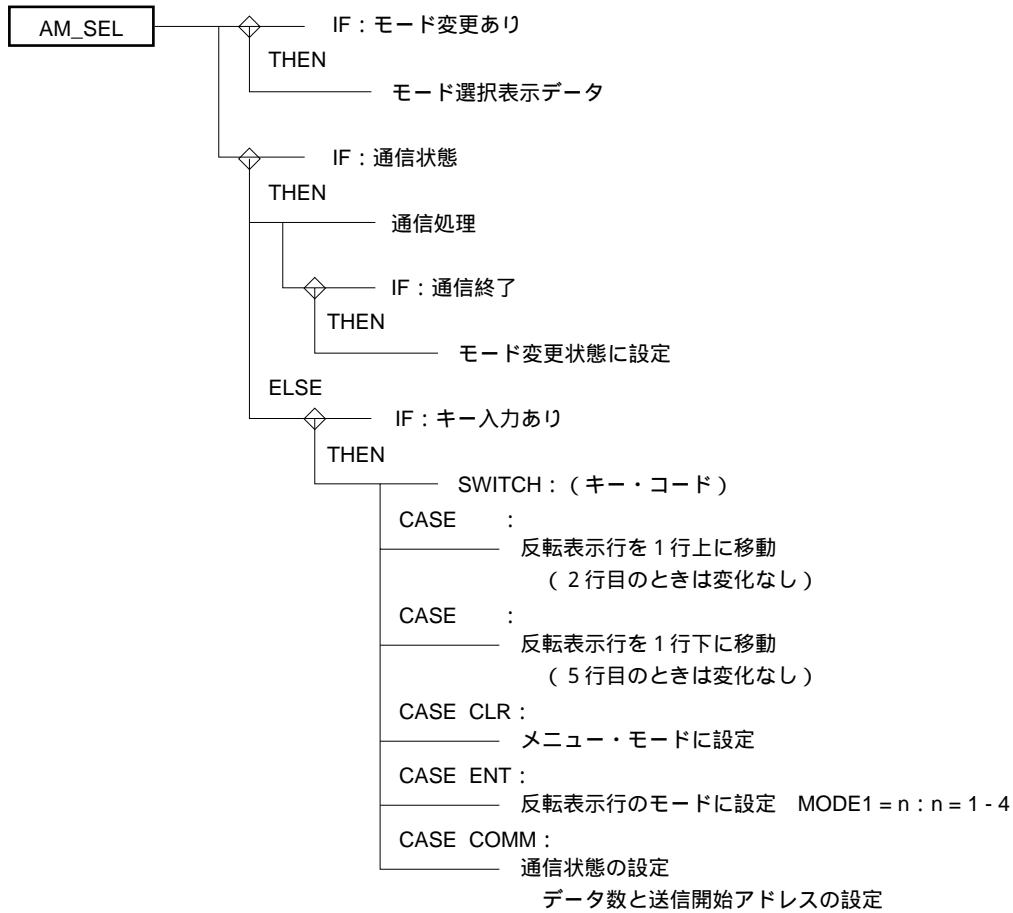
最終アドレス（BF00H-BFFFH）に空きデータ（FFH）を書き込みます。

4.4.8 SPDチャート

名刺(個人)モード

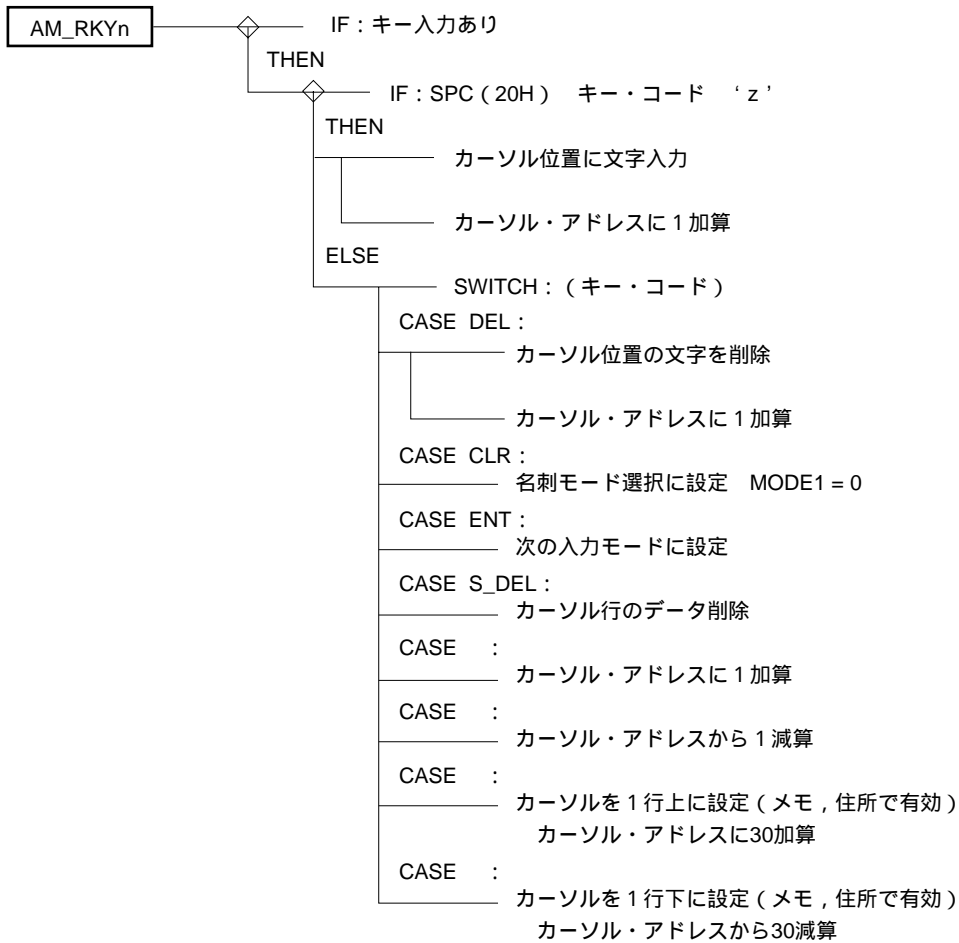
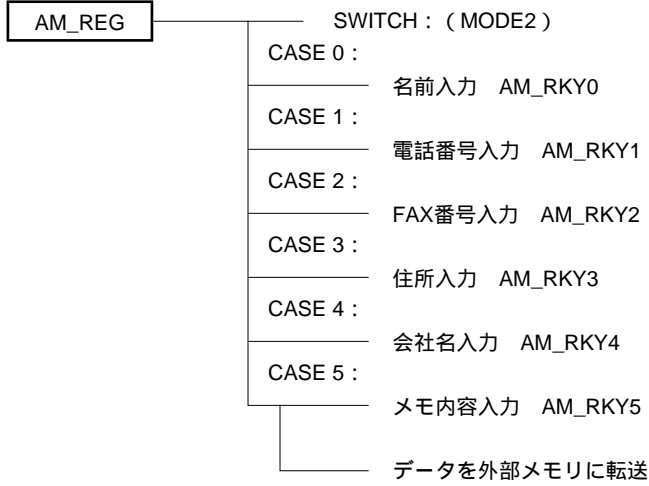


・モード選択



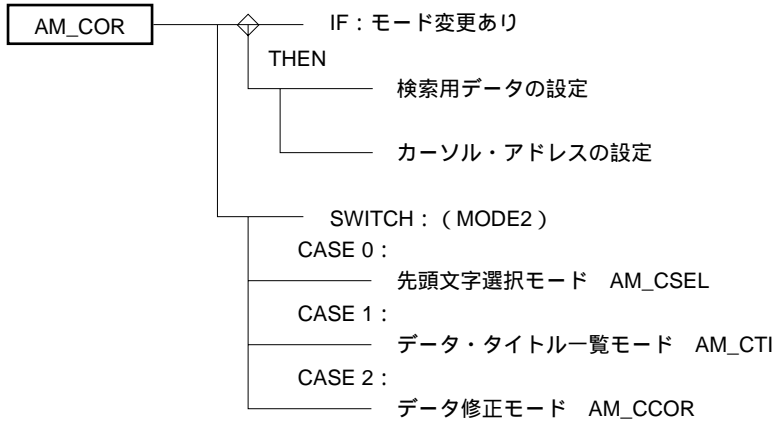
名刺（個人）モード

・登録モード



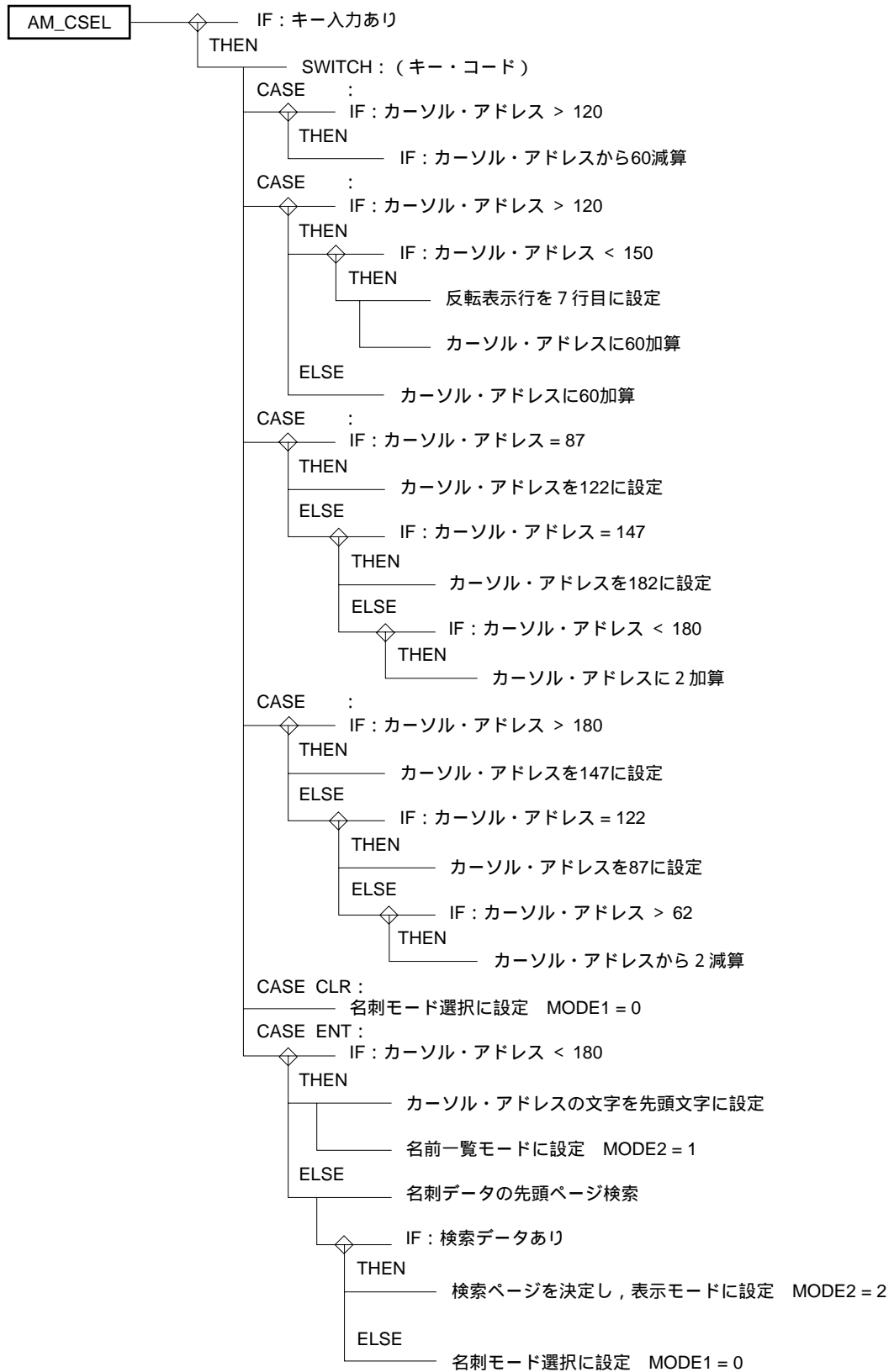
名刺(個人)モード

・修正モード



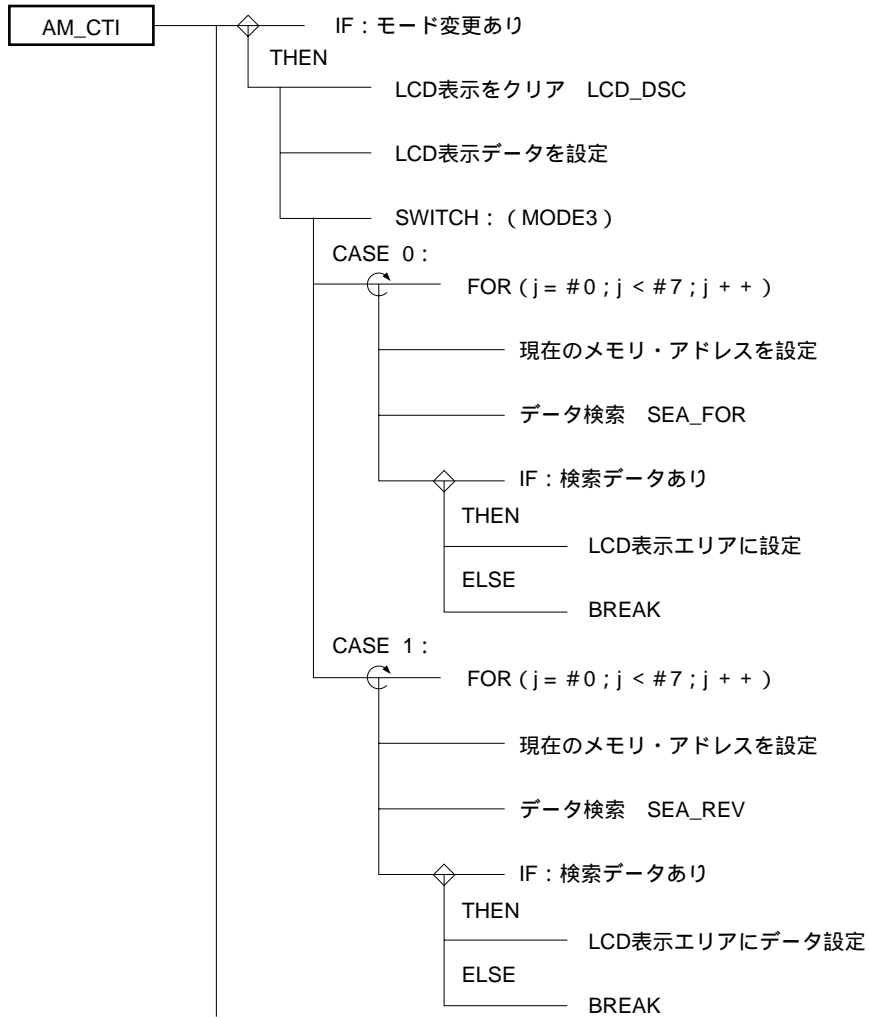
名刺(個人)モード

・修正モード



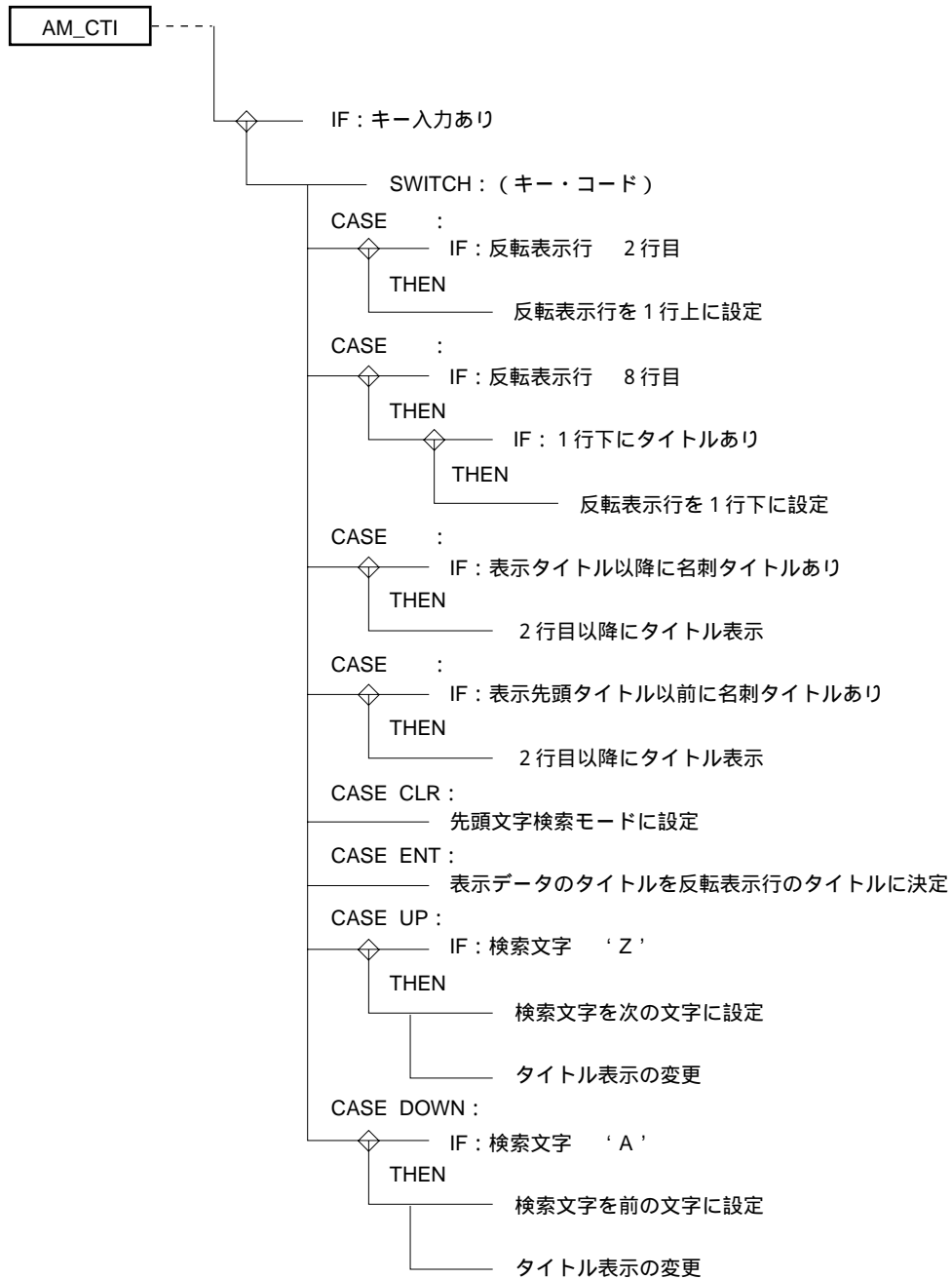
名刺(個人)モード

・修正モード



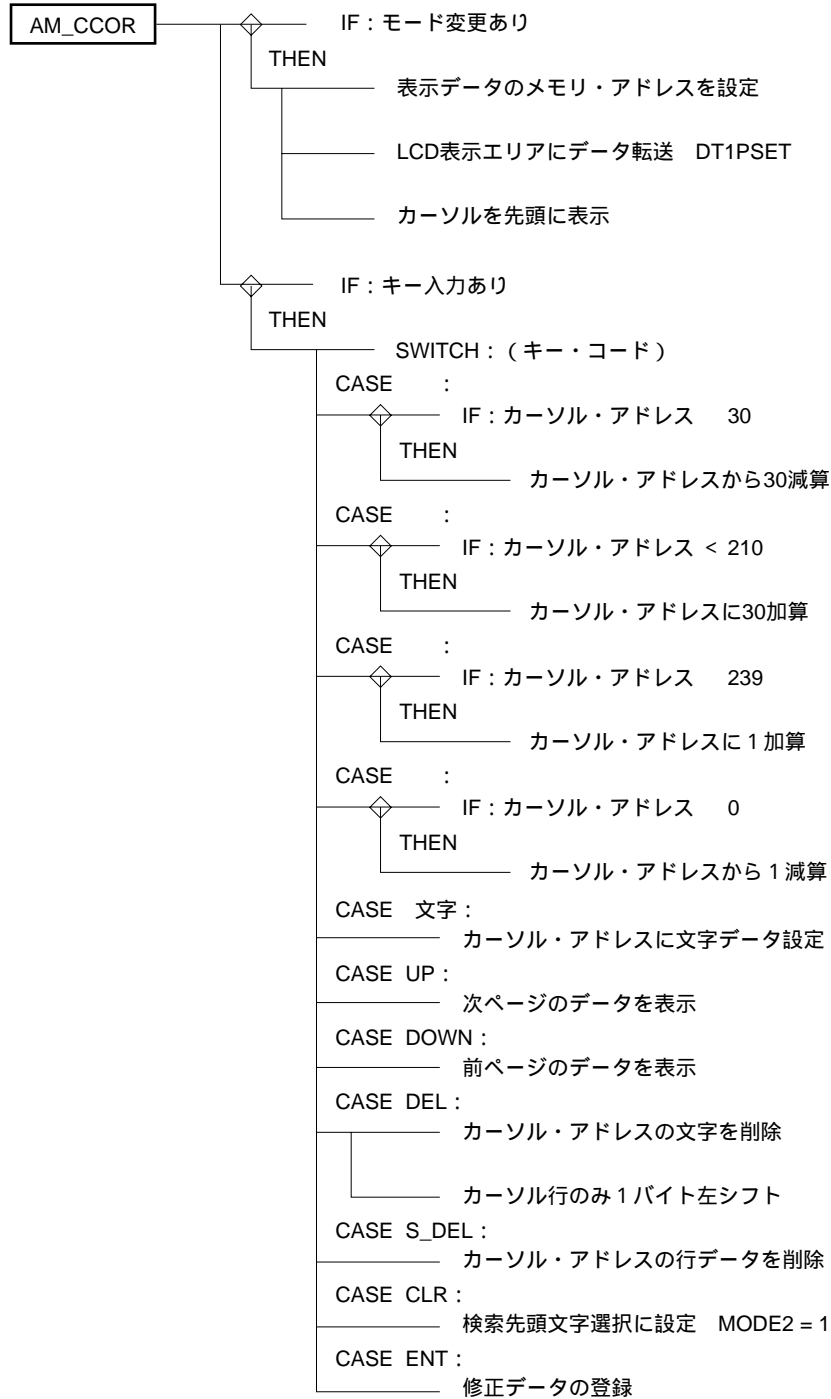
名刺(個人)モード

・修正モード



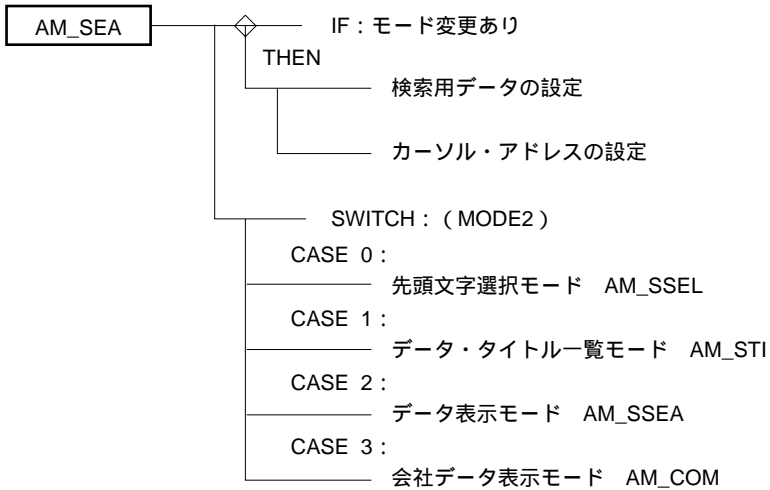
名刺(個人)モード

・修正モード



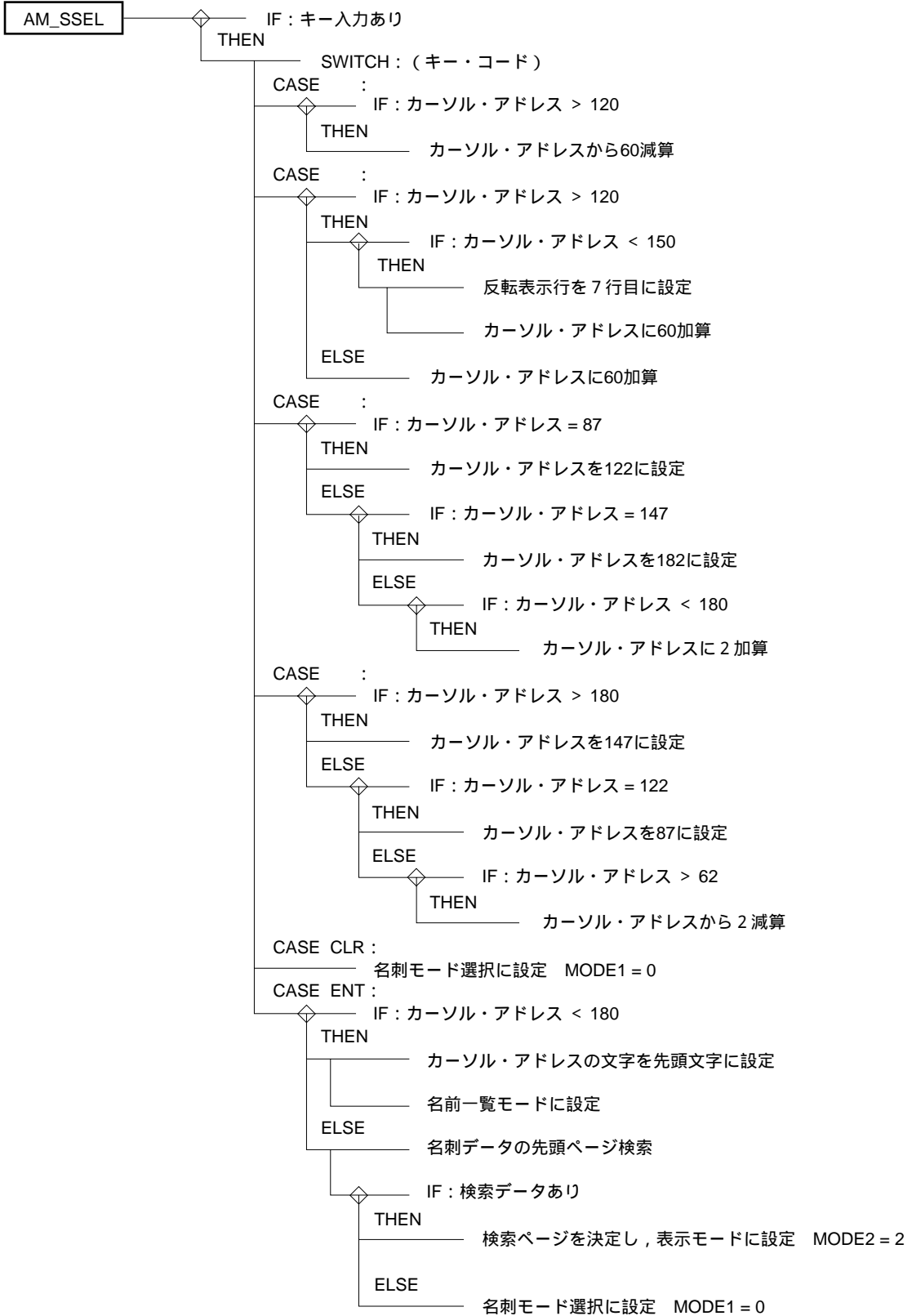
名刺（個人）モード

・検索モード



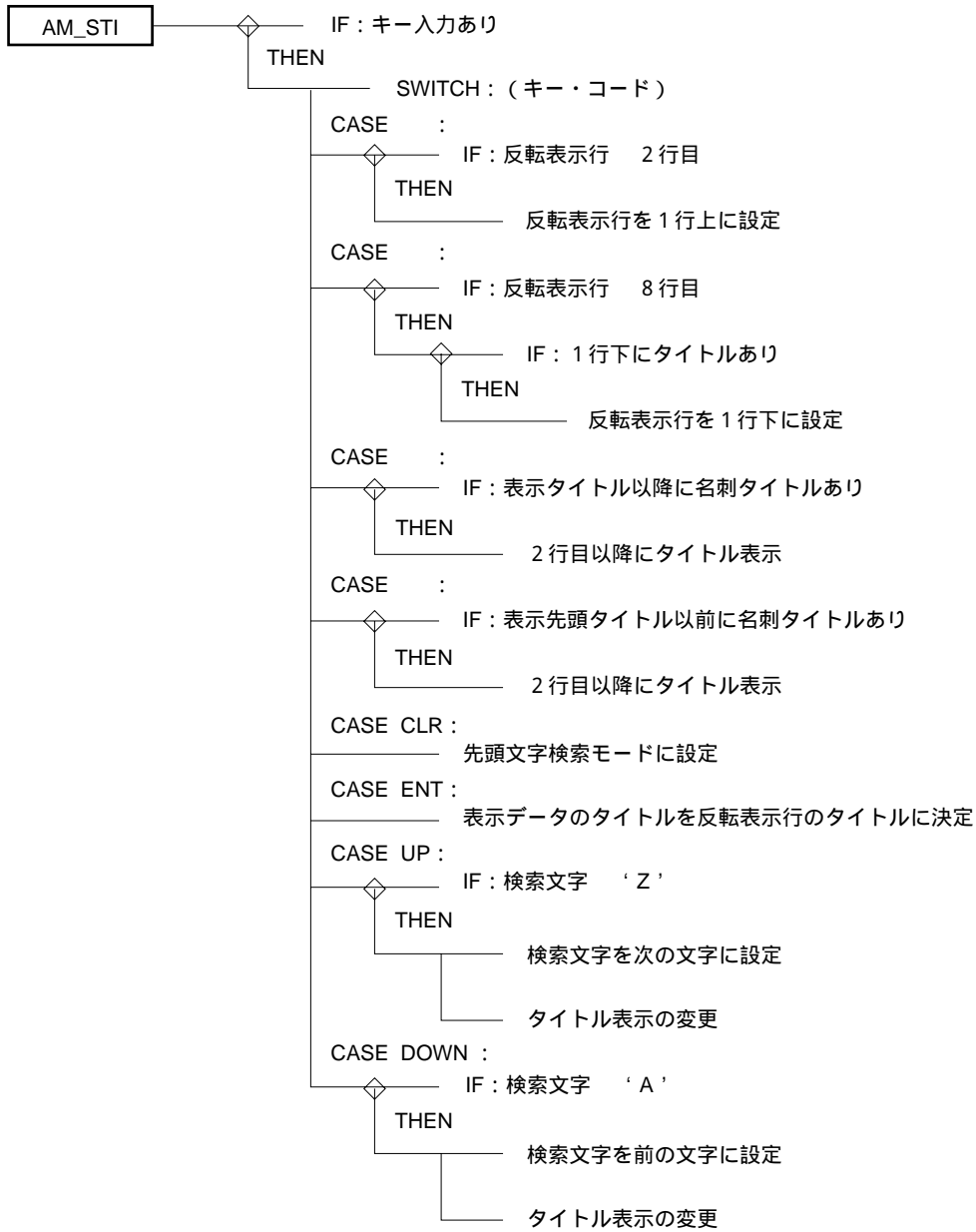
名刺(個人)モード

・検索モード



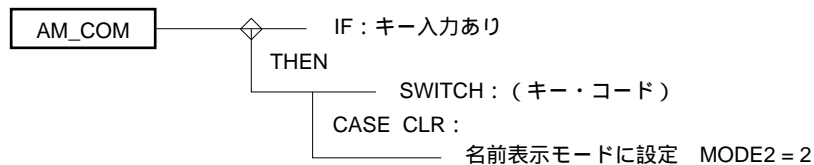
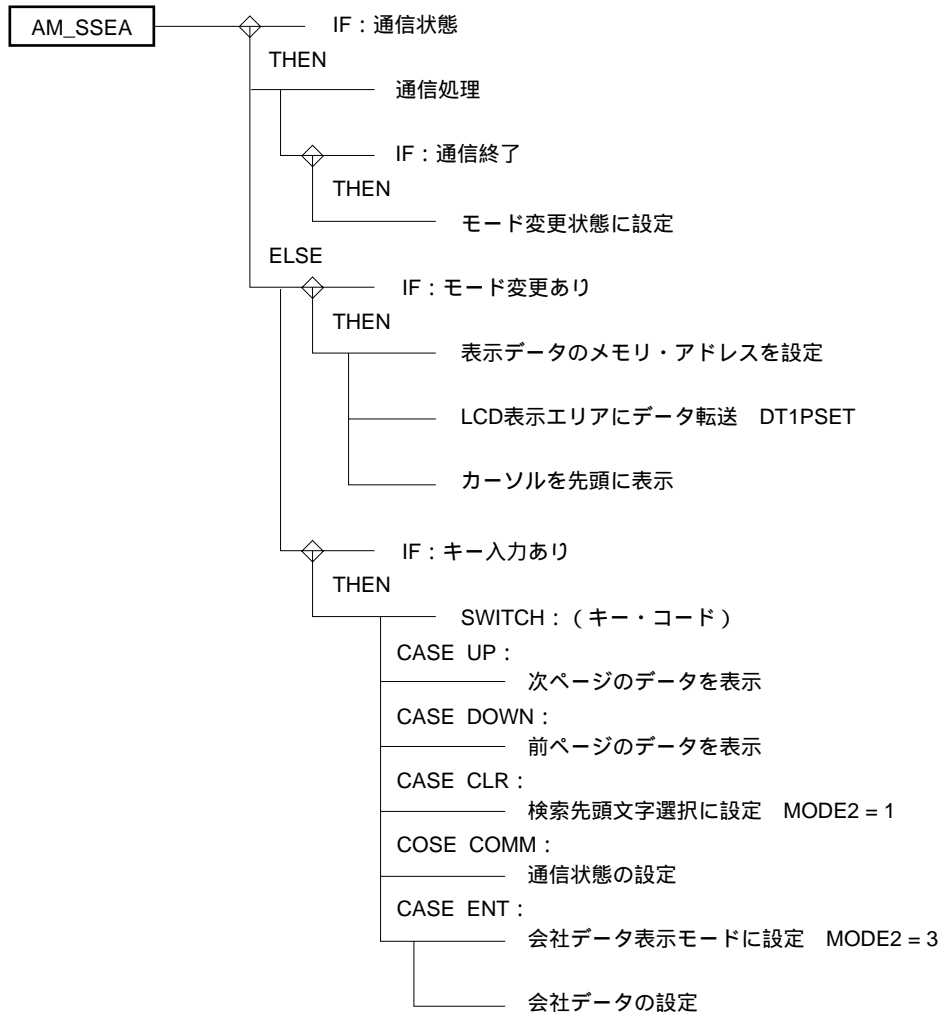
名刺（個人）モード

・検索モード



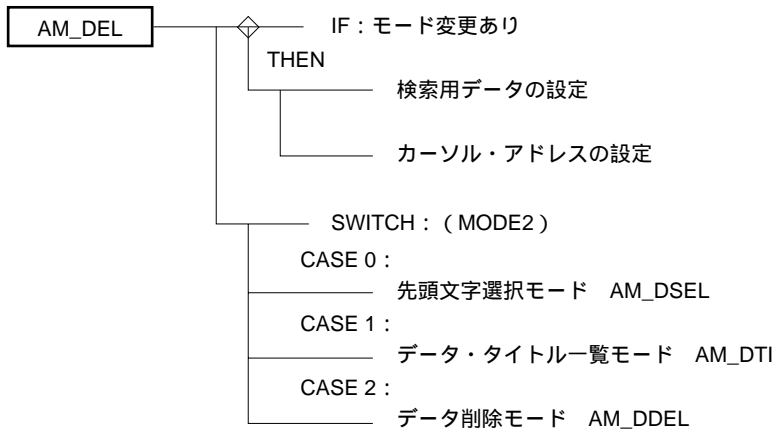
名刺(個人)モード

・検索モード



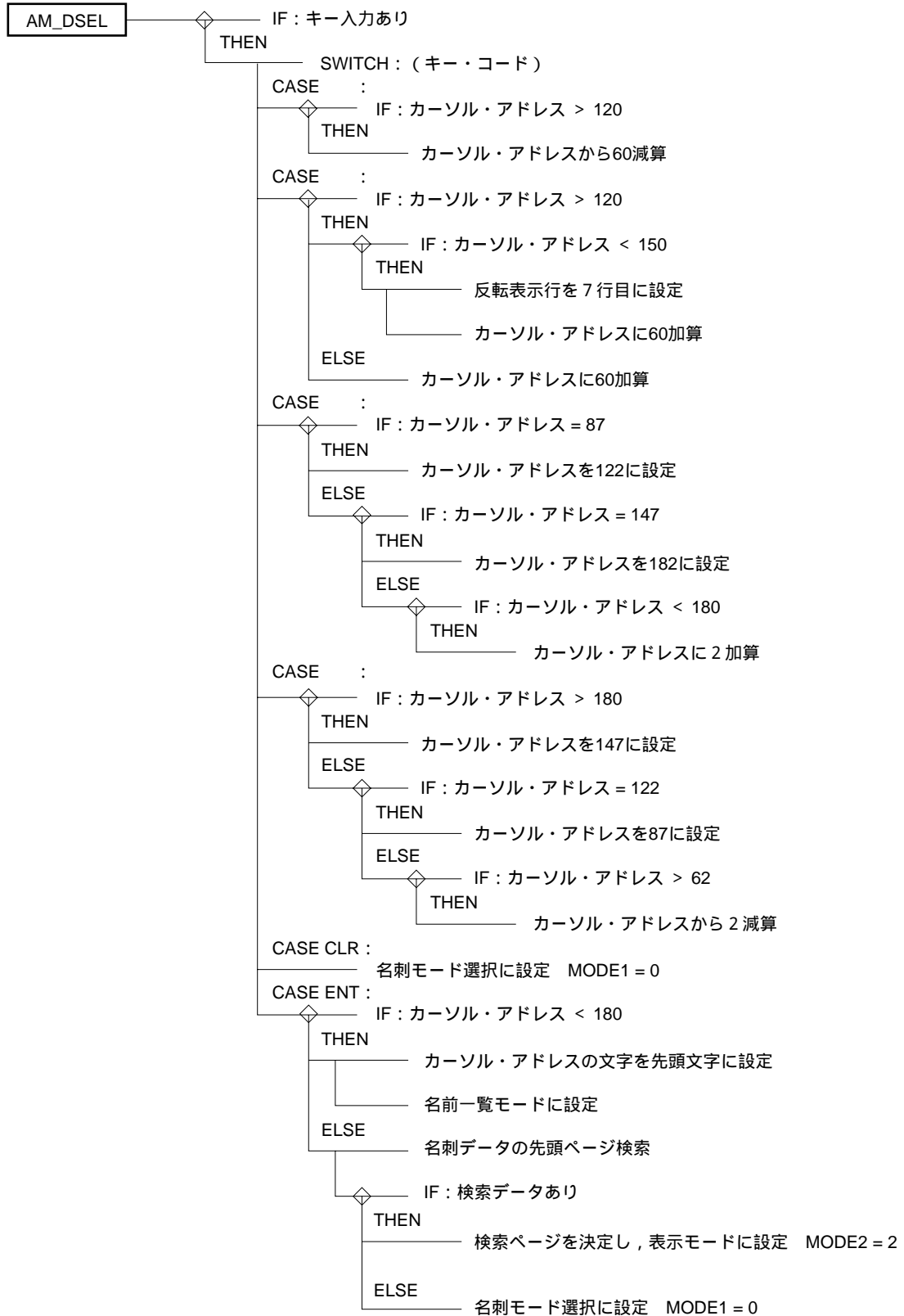
名刺(個人)モード

・削除モード



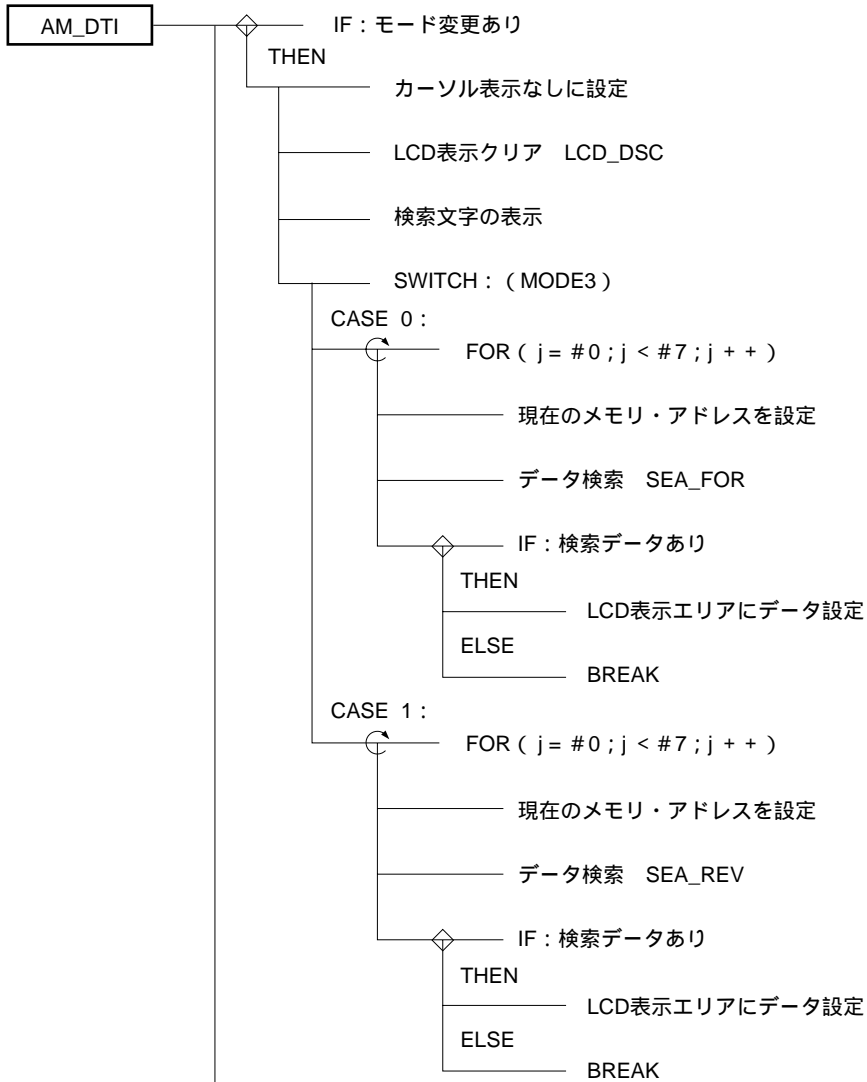
名刺（個人）モード

・削除モード



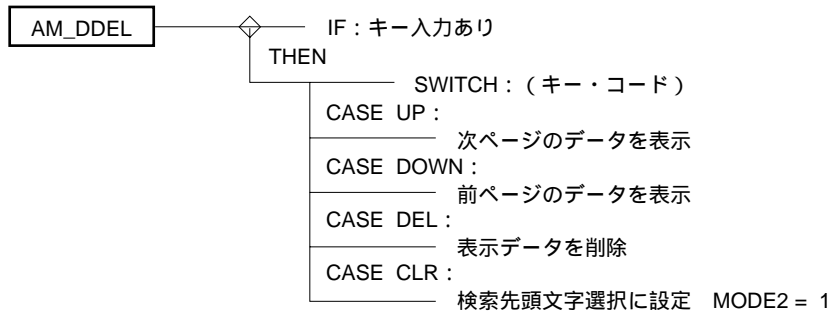
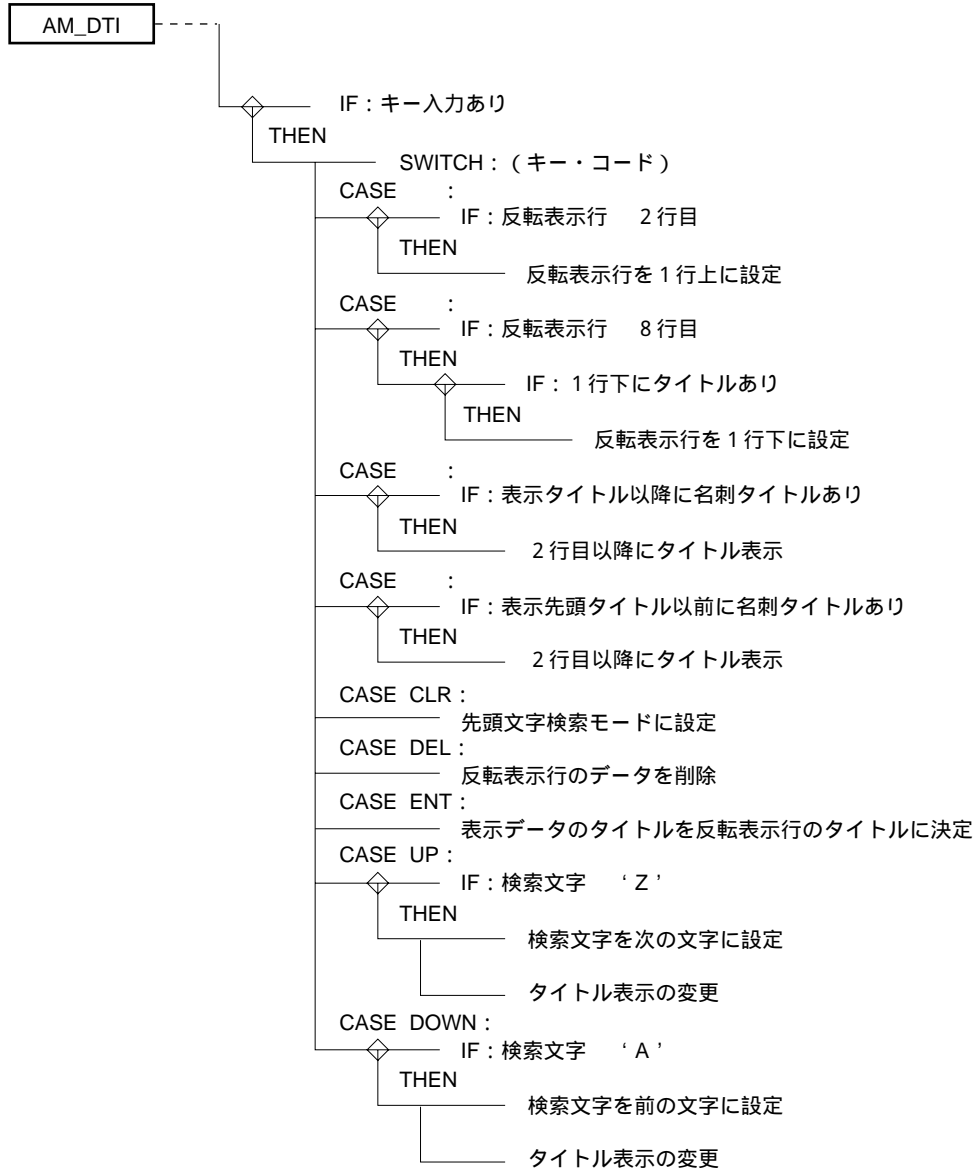
名刺(個人)モード

・削除モード



名刺(個人)モード

・削除モード



4.4.9 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
EXTRN  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, MENPAG, MEMBNK, BACPAG, BACBNK
EXTRN  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
EXTRN  COM_VOL, TRN_ADR, TRN_BNK, VAC_DAT, ADM_DAT, ADC_DAT, MEN_DAT, SCA_DAT, HOL_DAT
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DTILSET, DTIPSET, LCD_C, LCD_D, DT_SAV, STA_SEA, S_COMM
EXTRN  SAVADR, DTCOMP, DT_DEL, BANKSET, SEA_FOR, SEA_REV, ADR_INC, ADR_DEC, NEX_ADR, BEF_ADR
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, IV_CHGF, COMM_F, DS_ERAF
EXTBIT  CR_DSPF, ADENDF, ADSEAF, CMPERF, TR_CG1F, TR_CG2F, TR_CG3F, LCDCHGF

PUBLIC  S_ADMAN, ADM_VOL

$      INCLUDE (LCDCMD. INC)
$      INCLUDE (LCDT_EX. INC)

$LIST

      DSEG      SADDR
ADM_VOL:DS      1

;*****
;      名刺 (個人) 分岐処理
;*****
ADMSEG  CSEG
S_ADMAN:
      switch(MODE1)
      case 0:
            CALL  !AM_SEL          ;モード選択
            break
      case 1:
            CALL  !AM_REG          ;データ登録
            break
      case 2:
            CALL  !AM_COR          ;データ修正
            break
      case 3:
            CALL  !AM_SEA          ;データ検索
            break
      case 4:
            CALL  !AM_DEL          ;データ削除
      ends
      RET

;*****
;      名刺 (個人) モード選択処理
;*****
AM_SEL:
      if_bit(MD_CG1F)
            CLRI  MD_CG1F
            HL=#LCDDS          ;メニュー表示データの設定
            DE=#ADM_D          ;LCD 1行目のデータを設定
            CALL  !DTILSET

            DE=#REG_D          ;LCD 2行目のデータを設定
            CALL  !DTILSET

            DE=#COR_D          ;LCD 3行目のデータを設定
            CALL  !DTILSET

            DE=#SEA_D          ;LCD 4行目のデータを設定
            CALL  !DTILSET

            DE=#DEL_D          ;LCD 5行目のデータを設定
            CALL  !DTILSET

            DE=#CLR1LD          ;LCD 6行目のデータを設定

```

```

CALL !DTILSET

DE=#CLRILD ;LCD 7行目のデータを設定
CALL !DTILSET

DE=#CLRILD ;LCD 8行目のデータを設定
CALL !DTILSET

MODE2=#0
MODE3=#0
MODE4=#0
SET1 MD_CG2F
SET1 MD_CG3F
SET1 MD_CG4F
CLR1 CR_DSPF
SET1 IV_CHGF ;反転表示行を2行目に設定
INV_DS=#2
SET1 LCDCHGF
endif

if_bit(COMM_F) ;通信状態
CALL !S_COMM ;通信処理
if_bit(!COMM_F) ;通信終了
SET1 MD_CG1F ;通信前の状態に設定
endif
else
if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD) ;キー・コード分岐
case CUR_UP:
if(INV_DS>#2)
INV_DS--
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CUR_DN: ;反転表示行を下に移動
if(INV_DS<#5)
INV_DS++
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CLEAR:
MODE0=#0 ;前画面モードに変更
SET1 MD_CG0F
break
case ENTER: ;反転表示行のモード実行
A=INV_DS
A--
MODE1=A
SET1 MD_CG1F ;モード変更状態に設定
break
case COMM:
COM_VOL=ADM_VOL (A) ;通信状態に設定
B=#ADM_DAT ;送信データ数の設定
CALL !STA_SEA ;検索データの設定
TRN_ADR=H (A) ;送信データ数の設定
TRN_BNK=M_BANK (A)
SET1 COMM_F
SET1 TR_CG1F
ends
endif
endif
RET

;*****
; 名刺(個人)データ登録処理
;*****
AM_REG:
switch(MODE2)
case 0:
if_bit(MD_CG2F)
CLR1 MD_CG2F
INV_DS=#0 ;反転表示行削除
SET1 IV_CHGF

```

```

CALL !LCD_DSC
HL=#LCDDS                                ;カーソル・アドレスの設定
DE=#NAME_D
CALL !DTILSET                              ;0000H(1行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY0
break
case 1:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+30                              ;カーソル・アドレスの設定
DE=#TEL_D
CALL !DTILSET                              ;0000H+30(2行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY1
break
case 2:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+60                              ;カーソル・アドレスの設定
DE=#FAX_D
CALL !DTILSET                              ;0000H+60(3行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY2
break
case 3:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+90                              ;カーソル・アドレスの設定
DE=#ADR_D
CALL !DTILSET                              ;0000H+90(4行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY3
break
case 4:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+150                             ;カーソル・アドレスの設定
DE=#COMP_D
CALL !DTILSET                              ;0000H+150(6行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY4
break
case 5:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+180                             ;カーソル・アドレスの設定
DE=#MEMO_D
CALL !DTILSET                              ;0000H+180(7行目)に設定
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AM_RKY5
break
case 6:
if_bit(MD_CG2F)
CLR1 MD_CG2F
CALL !LCD_DSC
HL=#LCDDS+90                              ;カーソル・アドレスの設定
DE=#MEN_OV                               ;エラー表示
CALL !DTILSET                              ;0000H+90(4行目)に設定
SETI LCDCHGF
endif
CALL !AM_RKY6
ends

```

```

RET

;*****
; 名刺(個人)データ登録キー入力処理
;*****
AM_RKYO:
IF_BIT(KY_SIGF)                ;キー入力あり
CLR1 KY_SIGF
if_bit(DS_ERAF)                ;1ライン・データ削除
CLR1 DS_ERAF
HL=#LCDDS                      ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET                  ;0000H(1行目)に設定
CUR_ADR=#0
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
D=#30
CALL !AM_RDEL
break
case CLEAR:                    ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER:                    ;電話入力モードに設定
MODE2=#1
SET1 MD_CG2F
CLR1 CR_DSPF
break
case S_DEL:                    ;1ライン・データ削除
HL=#LCDDS                      ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET
CUR_ADR=#0                    ;カーソル・アドレスを先頭に設定
break
case CUR_RI:
if(CUR_ADR<#29)              ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
break
case CUR_LE:
if(CUR_ADR>#0)              ;カーソル・アドレス1行目先頭アドレス以上
CUR_ADR--
endif
ends
ELSE
B=CUR_ADR(A)                ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD(A)            ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#29)            ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
ENDIF
RET

AM_RKY1:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
CLR1 DS_ERAF
HL=#LCDDS+30                ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET                ;0000H+30(2行目)に設定
CUR_ADR=#30
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z')
switch(KEY_CD)
case DELETE:
D=#60
CALL !AM_RDEL
break

```

```

case CLEAR:                                ;前モードに設定
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:                                ;FAX入力モードに設定
    MODE2=#2
    SET1 MD_CG2F
    CLR1 CR_DSPF
    break
case S_DEL:                                ;1ライン・データ削除
    HL=#LCDDS+30                            ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL !DT1LSET
    CUR_ADR=#30                            ;カーソル・アドレスを先頭に設定
    break
case CUR_R1:
    if(CUR_ADR<#59)                        ;カーソル・アドレス2行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#30)                        ;カーソル・アドレス2行目先頭アドレス以上
        CUR_ADR--
    endif
ends
ELSE
B=CUR_ADR(A)                              ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD(A)                          ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#59)                            ;カーソル・アドレス2行目最終行以内
    CUR_ADR++
endif
ENDIF
ENDIF
RET

AM_RKY2:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
CLR1 DS_ERAF
HL=#LCDDS+60                              ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET                              ;0000H+60(3行目)に設定
CUR_ADR=#60
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#z')
switch(KEY_CD)
case DELETE:
D=#90
CALL !AM_RDEL
break
case CLEAR:                                ;前モードに設定
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:                                ;住所入力モードに設定
    MODE2=#3
    SET1 MD_CG2F
    CLR1 CR_DSPF
    break
case S_DEL:                                ;1ライン・データ削除
    HL=#LCDDS+60                            ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL !DT1LSET
    CUR_ADR=#80                            ;カーソル・アドレスを先頭に設定
    break
case CUR_R1:
    if(CUR_ADR<#89)                        ;カーソル・アドレス3行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#60)                        ;カーソル・アドレス3行目先頭アドレス以上

```

```

        CUR_ADR--
    endif
ends
ELSE
    B=CUR_ADR (A)           ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)       ;カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#89)         ;カーソル・アドレス3行目最終行以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

AM_RKYS:
IF_BIT(KY_SIGF)
    CLR1 KY_SIGF
    if_bit(DS_ERAF)
        CLR1 DS_ERAF
        HL=#LCDDS+90       ;カーソル・アドレスの設定
        DE=#CLR1LD
        CALL !DT1LSET      ;0000H+90(4行目)に設定
        CUR_ADR=#90
        SET1 CR_DSPF
    endif
    IF(KEY_CD<#SPC || KEY_CD>#'z')
        switch(KEY_CD)
        case DELETE:
            D=#150
            CALL !AM_RDEL
            break
        case CLEAR:       ;前モードに設定
            MODE1=#0
            SET1 MD_CG1F
            break
        case ENTER:      ;会社入力モードに設定
            MODE2=#4
            SET1 MD_CG2F
            CLR1 CR_DSPF
            break
        case S_DEL:      ;1ライン・データ削除
            if(CUR_ADR<#120)
                HL=#LCDDS+90 ;カーソル・アドレスの設定
                DE=#CLR1LD
                CALL !DT1LSET ;カーソル・アドレスを先順に設定
                CUR_ADR=#90
            else
                HL=#LCDDS+120 ;カーソル・アドレスの設定
                DE=#CLR1LD
                CALL !DT1LSET ;カーソル・アドレスを先順に設定
                CUR_ADR=#120
            endif
            break
        case CUR_RI:     ;カーソル・アドレス5行目最終行以内
            if(CUR_ADR<#149)
                CUR_ADR++
            endif
            break
        case CUR_LE:     ;カーソル・アドレス4行目先順アドレス以上
            if(CUR_ADR>#90)
                CUR_ADR--
            endif
            break
        case CUR_UP:     ;カーソル・アドレスを1行上(+30)に設定
            if(CUR_ADR>#120)
                CUR_ADR-=#30
            endif
            break
        case CUR_DN:     ;カーソル・アドレスを1行下(-30)に設定
            if(CUR_ADR<#120)
                CUR_ADR+=#30
            endif
        ends
    ELSE
        B=CUR_ADR (A)           ;カーソル・アドレスの設定

```



```

HL=#LCDDS
[HL+B]=KEY_CD (A) ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#149) ;カーソル・アドレス5行目最終行以内
  CUR_ADR++
endif
ENDIF
ENDIF
RET

AM_RKY4:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
  CLR1 DS_ERAF
  HL=#LCDDS+150 ;カーソル・アドレスの設定
  DE=#CLR1LD
  CALL !DTILSET ;0000H+150(8行目)に設定
  CUR_ADR=#150
  SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z')
  switch(KEY_CD)
  case DELETE:
    D=#180
    CALL !AM_RDEL
    break
  case CLEAR: ;前モードに設定
    MODE1=#0
    SET1 MD_CG1F
    break
  case ENTER: ;メモ入力モードに設定
    MODE2=#5
    SET1 MD_CG2F
    CLR1 CR_DSPF
    break
  case S_DEL: ;1ライン・データ削除
    HL=#LCDDS+150 ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL !DTILSET
    CUR_ADR=#150 ;カーソル・アドレスを先頭に設定
    break
  case CUR_RI:
    if(CUR_ADR<#179) ;カーソル・アドレス行目最終行以内
      CUR_ADR++
    endif
    break
  case CUR_LE:
    if(CUR_ADR>#150) ;カーソル・アドレス8行目先頭アドレス以上
      CUR_ADR--
    endif
  ends
ELSE
  B=CUR_ADR (A) ;カーソル・アドレスの設定
  HL=#LCDDS
  [HL+B]=KEY_CD (A) ;カーソル・アドレス位置に文字データを設定
  if(CUR_ADR<#179) ;カーソル・アドレス8行目最終行以内
    CUR_ADR++
  endif
ENDIF
ENDIF
RET

AM_RKY5:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
  CLR1 DS_ERAF
  HL=#LCDDS+180 ;カーソル・アドレスの設定
  DE=#CLR1LD
  CALL !DTILSET ;0000H+180(7行目)に設定
  CUR_ADR=#180
  SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z')
  switch(KEY_CD)

```

```

case DELETE:
    D=#240
    CALL !AM_RDEL
    break
case CLEAR:
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:
    M_BANK=#3
    CALL !BANKSET
    A=!0BF00H
    if(A=#VAC_DAT)
        CLR1 ADSEAF
        M_BANK=#0
        CALL !BANKSET
        H=#ADRSTA
        L=#0
        B=#ADM_DAT
        C=#16
        DE=#LCDDS
        CALL !SAVADR
        DE=#DTWORK
        [DE]=#ADM_DAT (A)
        DE++
        A=#0
        for(i=#1;i<#16;i++)
            [DE]=A
            DE++
        next
        HL=#LCDDS
        for(i=#0;i<#240;i++)
            [DE]=[HL] (A)
            DE++
            HL++
        next
        DE=#DTWORK
        H=MEMPAG (A)
        L=#0
        CALL !DT_SAV
        ADM_VOL++
        MODE1=#0
        SET1 MD_CG1F
    else
        SET1 MD_CG2F
        MODE2=#6
    endif
    CLR1 CR_DSPF
    break
case S_DEL:
    if(CUR_ADR<#210)
        HL=#LCDDS+180
        DE=#CLR1LD
        CALL !DT1LSET
        CUR_ADR=#180
    else
        HL=#LCDDS+210
        DE=#CLR1LD
        CALL !DT1LSET
        CUR_ADR=#210
    endif
    break
case CUR_RI:
    if(CUR_ADR<#239)
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#180)
        CUR_ADR--
    endif
    break
case CUR_UP:
    if(CUR_ADR>=#210)
        CUR_ADR-=#30
    endif

```

```

        endif
        break
    case CUR_DN:
        if(CUR_ADR<#210)
            CUR_ADR+=#30          ;カーソル・アドレスを1行下(-30)に設定
        endif
    ends
ELSE
    B=CUR_ADR (A)                ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)            ;カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#239)
        CUR_ADR+
    endif
ENDIF
ENDIF
RET

AM_RKY6:
    if_bit(KY_SIGF)
        CLR1 KY_SIGF
        switch(KEY_CD)
        case CLEAR:              ;前モードに設定
            MODE1=#0
            SET1 MD_CG1F
        ends
    endif
    RET

AM_RDEL:
    B=CUR_ADR (A)                ;現在のカーソル・アドレスを設定
    C=CUR_ADR (A)                ;データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(C<D) (A)
        [HL+B]=[HL+C] (A)        ;データ転送
        B++
        C++
    endw
    [HL+B]=#SPC (A)              ;行最終アドレスにスペース・データを設定
    RET

;*****
; 名刺(個人)修正モード
;*****
AM_COR:
    if_bit(MD_CG1F)
        CLR1 MD_CG1F
        INV_DS=#0                ;反転表示行削除
        SET1 IV_CHGF
        HL=#LCDDS                ;メニュー表示データの設定
        DE=#NA_SEA
        CALL !DTILSET            ;LCD1行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET            ;LCD2行目のデータを設定

        DE=#ATOM_D
        CALL !DTILSET            ;LCD3行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET            ;LCD4行目のデータを設定

        DE=#NTOZ_D
        CALL !DTILSET            ;LCD5行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET            ;LCD6行目のデータを設定

        DE=#NEXT_P
        CALL !DTILSET            ;LCD7行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET            ;LCD8行目のデータを設定

```

```

MODE2=#0
MODE3=#0
MODE4=#0

SET1 CR_DSPF           ;カーソル表示の設定
CUR_ADR=#62           ;カーソル・アドレスを'A'の位置に設定
MEMPAG=#ADRSTA
MEMBNK=#0
SET1 LCDCHGF
endif
switch(MODE2)
case 0:
CALL IAM_CSEL         ;先頭文字選択モード
break
case 1:
CALL IAM_CTI         ;名前選択モード
break
case 2:
CALL IAM_CCOR        ;データ修正モード
ends
RET

;*****
;  名刺(個人)先頭文字選択モード
;*****
AM_CSEL:
AM_SSEL:
AM_DSEL:
IF_BIT(KY_SIGF)      ;キー入力あり
CLR1 KY_SIGF
switch(KEY_CD)
case CUR_RI:
if(CUR_ADR==#86)    ;カーソル・アドレス=M
CUR_ADR=#122       ;カーソル・アドレス=N
elseif(CUR_ADR==#146) ;カーソル・アドレス=Z
CUR_ADR=#182       ;7行目反転表示
INV_DS=#7
SET1 IV_CHGF
elseif(CUR_ADR<#180) ;カーソル・アドレス上記以外の値
CUR_ADR+=#2        ;カーソル・アドレスを次の文字に設定
endif
break
case CUR_LE:
if(CUR_ADR>#180)    ;反転表示行
CUR_ADR=#146       ;カーソル・アドレス=Z
INV_DS=#0
SET1 IV_CHGF       ;反転表示消去
elseif(CUR_ADR==#122) ;カーソル・アドレス=N
CUR_ADR=#86        ;カーソル・アドレス=M
elseif(CUR_ADR>#62) ;カーソル・アドレス上記以外の値
CUR_ADR-=#2        ;カーソル・アドレスを前の文字に設定
endif
break
case CUR_UP:
if(CUR_ADR>#120)    ;カーソル・アドレスN以降
CUR_ADR-=#60       ;カーソル・アドレスを1行上(-60)に設定
INV_DS=#0
SET1 IV_CHGF       ;反転表示消去
endif
break
case CUR_DN:
if(CUR_ADR>#120)    ;カーソル・アドレスN以降
if(CUR_ADR<#150)    ;反転表示でない
INV_DS=#7
SET1 IV_CHGF       ;7行目反転表示
CUR_ADR+=#60       ;カーソル・アドレスを1行下(+60)に設定
endif
else
CUR_ADR+=#60       ;カーソル・アドレスを1行下(+60)に設定
endif
break
case CLEAR:         ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break

```

```

case ENTER:
  if(CUR_ADR<#180)
    HL=#LCDDS
    B=CUR_ADR (A)
    SEACHR=[HL+B] (A)
    SET1 MD_CG2F
    SET1 MD_CG3F
    MODE2=#1
    MODE3=#0
    BACPAG=MEMPAG (A)
    BACBNK=MEMBNK (A)
    CALL !LCD_DSC
  else
    B=#ADM_DAT
    CALL !STA_SEA
    if_bit(ADENDF)
      if([HL]==B) (A)
        MEMBNK=M_BANK (A)
        MEMPAG=M (A)
        MODE2=#2
        SET1 MD_CG2F
        SET1 MD_CG3F
      else
        MODE1=#0
        SET1 MD_CG1F
      endif
    else
      MODE1=#0
      SET1 MD_CG1F
    endif
  endif
endif
ends
ENDIF
RET

;*****
; 名刺 (個人) 名前選択モード
;*****
AM_CTI:
AM_STI:
  if_bit(MD_CG2F)
    CLR1 MD_CG2F
    CLR1 CR_DSPF
    CALL !NAM_DSP
    FOR_BNK=#0
    FOR_ADR=#ADRSTA
    REV_BNK=#0
    REV_ADR=#ADRSTA
    MODE3=#0
    SET1 MD_CG3F
  endif
  IF_BIT(MD_CG3F)
    CLR1 MD_CG3F
    SWITCH(MODE3)
    CASE 0:
      FOR(j=#0;j<#7;j++)
        H=FOR_ADR (A)
        L=#00H
        M_BANK=FOR_BNK (A)
        CALL !BANKSET
        B=#ADM_DAT
        C=#16
        CALL !SEA_FOR
        if_bit(ADSEAF)
          D=H (A)
          E=#16
          PUSH DE
          switch(j)
          case 0:
            CALL !MEX_ADR
            CALL !NAM_DSP
            HL=#LCDDS+30
            break
          case 1:

```

```

        HL=#LCDDS+60
        break
    case 2:                                ; 4行目に設定
        HL=#LCDDS+90
        break
    case 3:                                ; 5行目に設定
        HL=#LCDDS+120
        break
    case 4:                                ; 6行目に設定
        HL=#LCDDS+150
        break
    case 5:                                ; 7行目に設定
        HL=#LCDDS+180
        break
    case 6:                                ; 8行目に設定
        HL=#LCDDS+210
    ends
    POP    DE
    CALL  !DTILSET
    H=D (A)
    CALL  !ADR_INC
    FOR_ADR=H (A)
    FOR_BNK=M_BANK (A)
elseif_bit(ADENDF)                        ; 検索データなし
    break
endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)                        ; 1ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)                    ; 現在のメモリ・アドレスを設定
    CALL  !BANKSET
    B=#ADM_DAT
    C=#16
    CALL  !SEA_REV
    if_bit(ADSEAF)
        D=H (A)                            ; 転送元アドレスの設定
        E=#16
        PUSH DE                            ; アドレス通過
        switch(j)                          ; 転送先アドレスの設定
        case 0:
            CALL  !BEF_ADR
            CALL  !NAM_DSP
            HL=#LCDDS+210
            break
        case 1:                            ; 7行目に設定
            HL=#LCDDS+180
            break
        case 2:                            ; 6行目に設定
            HL=#LCDDS+150
            break
        case 3:                            ; 5行目に設定
            HL=#LCDDS+120
            break
        case 4:                            ; 4行目に設定
            HL=#LCDDS+90
            break
        case 5:                            ; 3行目に設定
            HL=#LCDDS+60
            break
        case 6:                            ; 2行目に設定
            HL=#LCDDS+30
        ends
        POP    DE
        CALL  !DTILSET
        H=D (A)
        CALL  !ADR_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
elseif_bit(ADENDF)                        ; 検索データなし
    break
endif
NEXT

```

```

ENDS
INV_DS=#2
SET1  IV_CHGF
ENDIF

IF_BIT(KY_SIGF)                                ;キー入力あり
CLR1  KY_SIGF
switch(KEY_CD)
case CUR_UP:                                   ;反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SET1  IV_CHGF                        ;反転表示行変更状態に設定
    endif
    break
case CUR_DN:                                   ;反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)                       ;転送先アドレスの設定
        case 2:                               ;3行目に設定
            HL=#LCDDS+80
            break
        case 3:                               ;4行目に設定
            HL=#LCDDS+90
            break
        case 4:                               ;5行目に設定
            HL=#LCDDS+120
            break
        case 5:                               ;6行目に設定
            HL=#LCDDS+150
            break
        case 6:                               ;7行目に設定
            HL=#LCDDS+180
            break
        case 7:                               ;8行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!="#SPC) (A)                   ;次の行に名前あり
            INV_DS++
            SET1  IV_CHGF                    ;反転表示行変更状態に設定
        endif
    endif
    break
case CUR_RI:
    MODE3=#1
    SET1  MD_CG3F
    break
case CUR_LE:
    MODE3=#0
    SET1  MD_CG3F
    break
case CLEAR:
    MODE2=#0
    SET1  MD_CG1F
    break
case ENTER:
    CLR1  ADSEAF
    M_BANK=#0                                ;外部メモリを開始位置に設定
    CALL  !BANKSET
    H=#ADRSTA
    L=#0
    B=#ADM_DAT                              ;比較データを名刺(個人)データに設定
    C=#18                                    ;1ライン比較の開始アドレスを設定(HL+C)
    switch(INV_DS)                            ;検索対象行アドレスの設定
    case 2:                                   ;2行目に設定
        DE=#LCDDS+30
        break
    case 3:                                   ;3行目に設定
        DE=#LCDDS+60
        break
    case 4:                                   ;4行目に設定
        DE=#LCDDS+90
        break
    case 5:                                   ;5行目に設定
        DE=#LCDDS+120
        break
    case 6:                                   ;6行目に設定

```

```

        DE=#LCDDS+150
        break
    case 7:                                ; 7行目に設定
        DE=#LCDDS+180
        break
    case 8:                                ; 8行目に設定
        DE=#LCDDS+210
    ends

    if([DE]!="#SPC) (A)
        repeat
            CALL !SAVADR                ;メモリ・アドレスを設定
            if_bit(CY)
            CALL !ADR_INC
            endif
            if([HL]==#VAC_DAT) (A)      ;検索アドレスのデータは空きデータ
                break                    ;登録データなし→検索終了
            else
                MODE2=#2                ;登録データあり→データ表示
                SET1 MD_CG3F
            endif
            until_bit(ICMPERF)

            MEMBNK=M_BANK (A)
            MENPAG=H (A)
        endif
        break
    case UP:
        if(SEACHR<#'Z')
            SET1 MD_CG2F                ;検索文字'A'-'Y'
            SEACHR++                    ;次の文字に変更
        endif
        break
    case DOWN:
        if(SEACHR>#'A')
            SET1 MD_CG2F                ;検索文字'B'-'Z'
            SEACHR--                    ;前の文字に変更
        endif
    ends
ENDIF
RET

;*****
; 名刺 (個人) 名前修正モード
;*****
AM_CCOR:
    if_bit(MD_CG3F)                    ;修正データの変更
        CLR1 MD_CG3F
        M_BANK=MENBNK (A)              ;メモリ・アドレスの設定
        CALL !BANKSET
        D=MENPAG (A)
        E=#16
        HL=#LCDDS                      ;LCD表示アドレスの設定
        CALL !DTIPSET                  ;表示データの転送
        CUR_ADR=#30                    ;カーソル・アドレスを1文字目に設定
        SET1 CR_DSPF
        INV_DS=#0
        SET1 IV_CHGF
        SET1 LCDCHGF
    endif
    IF_BIT(KY_SIGF)
        CLR1 KY_SIGF
        IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
            switch(KEY_CD)
            case DELETE:
                B=CUR_ADR (A)           ;現在のカーソル・アドレスを設定
                X=CUR_ADR (A)           ;桁数の設定
                A=#0
                C=#30                   ;1行の文字数
                AX/=C
                X=C (A)                 ;桁数
                C=CUR_ADR (A)           ;データ転送対象のアドレス(+1)を設定
                C++
                HL=#LCDDS
                while(X<#29) (A)

```



```

        [HL+B]=[HL+C] (A)      ;データ転送
        B++
        C++
        X++
    endw
    [HL+B]=#SPC (A)           ;行最終アドレスにスペース・データを設定
    break
case CLEAR:                  ;前モードに設定
    MODE2=#0
    SETI  MD_CG1F
    break
case ENTER:                  ;修正データの登録
    M_BANK=MEMBNK (A)
    CALL  !BANKSET
    H=MEMPAG (A)
    L=#16
    DE=#LCDDS
    CALL  !DTIPSET
    MODE1=#0
    SETI  MD_CG1F
    break
case S_DEL:                  ;1ライン・データ削除
    X=CUR_ADR (A)            ;カーソル・アドレスの取り込み
    A=#0H
    C=#30
    AX/=C                    ;1行文字数
    A=#30                    ;行数算出(X)
    AX*=X                    ;行先頭アドレスの設定
    AX+=#LCDDS
    HL=AX                    ;カーソル・アドレスの設定
    DE=#CLR!LD
    CALL  !DTILSET
    CUR_ADR=X (A)           ;カーソル・アドレスを先頭に設定
    break
case CUR_RI:
    if(CUR_ADR<#239)        ;カーソル・アドレス1行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#30)         ;カーソル・アドレス1行目先頭アドレス以上
        CUR_ADR--
    endif
    break
case CUR_UP:
    if(CUR_ADR>=#60)
        CUR_ADR-=#30        ;カーソル・アドレスを1行上(-30)に設定
    endif
    break
case CUR_DN:
    if(CUR_ADR<#210)
        CUR_ADR+=#30        ;カーソル・アドレスを1行下(+30)に設定
    endif
    break
case UP:
    M_BANK=MEMBNK (A)       ;メモリ・アドレスの設定
    CALL  !BANKSET
    H=MEMPAG (A)           ;表示データのページ設定
    L=#0
    if(H=#ADREND) (A)      ;バンク最終アドレス
        if(M_BANK<#3)      ;バンク0-バンク2
            M_BANK++        ;バンク更新
            CALL  !BANKSET
            H=#ADRSTA
            if([HL]==#ADM_DAT) (A) ;次データ=名刺(個人)データ
                SETI  MD_CG3F
                MEMPAG=H (A) ;アドレスの更新
                MEMBNK=M_BANK (A)
            endif
        endif
    else
        H++                ;次データ参照
        if([HL]==#ADM_DAT) (A)
            SETI  MD_CG3F    ;アドレスの更新
            MEMPAG=H (A)

```

```

MEMBNK=M_BANK (A)
endif
endif
break
case DOWN:
M_BANK=MEMBNK (A)           ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG (A)                ;表示データのページ設定
L=#0
if(H==#ADRSTA) (A)          ;バンク先頭アドレス
if(M_BANK)#0                ;バンク1-バンク3
M_BANK--                    ;バンク更新
CALL !BANKSET
H=#ADREND
if([HL]==#ADM_DAT) (A) ;次データ=名刺(個人)データ
SETI MD_CG3F
MEMPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H--                          ;次データ参照
if([HL]==#ADM_DAT) (A)
SETI MD_CG3F ;アドレスの更新
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
ends
ELSE
B=CUR_ADR (A)                ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)            ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#239)            ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
ENDIF
RET

;*****
; 名刺(個人)データ検索モード
;*****
AM_SEA:
if_bit(MD_CG1F)
CLR! MD_CG1F
INV_DS=#0                    ;反転表示行削除
SETI IV_CHGF
HL=#LCDDS                    ;メニュー表示データの設定
DE=#NA_SEA
CALL !DTILSET                ;LCD1行目のデータを設定

DE=#CLR1D
CALL !DTILSET                ;LCD2行目のデータを設定

DE=#ATOM_D
CALL !DTILSET                ;LCD3行目のデータを設定

DE=#CLR1D
CALL !DTILSET                ;LCD4行目のデータを設定

DE=#NTOZ_D
CALL !DTILSET                ;LCD5行目のデータを設定

DE=#CLR1D
CALL !DTILSET                ;LCD6行目のデータを設定

DE=#NEXT_P
CALL !DTILSET                ;LCD7行目のデータを設定

DE=#CLR1D
CALL !DTILSET                ;LCD8行目のデータを設定

MODE2=#0
MODE3=#0

```

```

MODE4=#0

SET1  CR_DSPF          ;カーソル表示の設定
CUR_ADR=#62          ;カーソル・アドレスを'A'の位置に設定
SET1  LCDCHGF

endif
switch(MODE2)
case 0:
CALL  !AM_SSEL        ;先頭文字選択モード
break
case 1:
CALL  !AM_STI        ;名前選択モード
break
case 2:
CALL  !AM_SSEA       ;データ表示モード
break
case 3:
CALL  !AM_COM        ;会社データ表示
ends
RET

;*****
:   名刺(個人)名前検索モード
;*****
AM_SSEA:
if_bit(COMM_F)       ;通信状態
CALL  !S_COMM       ;通信処理
if_bit(!COMM_F)     ;通信終了
SET1  MD_CG3F       ;通信前の状態に設定
endif
else
if_bit(MD_CG3F)     ;修正データの変更
CLR1  MD_CG3F
M_BANK=MEMBNK(A)   ;メモリ・アドレスの設定
CALL  !BANKSET
D=MEMPAG(A)
E=#16
HL=#LCDDS          ;LCD表示アドレスの設定
CALL  !DTIPSET     ;表示データの転送
CLR1  CR_DSPF     ;カーソル削除
INV_DS=#0          ;反転表示削除
SET1  IV_CHGF
SET1  LCDCHGF
endif
IF_BIT(KY_SIGF)
CLR1  KY_SIGF
switch(KEY_CD)
case CLEAR:
SET1  MD_CG1F     ;前モードに設定
break
case UP:
M_BANK=MEMBNK(A)   ;メモリ・アドレスの設定
CALL  !BANKSET
H=MEMPAG(A)        ;表示データのページ設定
L=#0
if(H==#ADREND)(A) ;バンク最終アドレス
if(M_BANK<#3)     ;バンク0-バンク2
M_BANK++         ;バンク更新
CALL  !BANKSET
H=#ADRSTA
if([HL]==#ADM_DAT)(A) ;次データ=名刺(個人)データ
SET1  MD_CG3F
MEMPAG=H(A)       ;アドレスの更新
MEMBNK=M_BANK(A)
endif
endif
else
H++               ;次データ参照
if([HL]==#ADM_DAT)(A)
SET1  MD_CG3F     ;アドレスの更新
MEMPAG=H(A)
MEMBNK=M_BANK(A)
endif
endif
break

```

```

case DOWN:
M_BANK=MEMBNK (A)           ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG (A)                 ;表示データのページ設定
L=#0
if(H=#ADRSTA) (A)           ;バンク先順アドレス
if(M_BANK>#0)               ;バンク0-バンク2
M_BANK--                     ;バンク更新
CALL !BANKSET
H=#ADREND
if([HL]==#ADM_DAT) (A)      ;次データ=名刺(個人)データ
SETI MD_CG3F
MEMPAG=H (A)                ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H--                           ;次データ参照
if([HL]==#ADM_DAT) (A)
SETI MD_CG3F                ;アドレスの更新
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
break
case COMM:
TRN_ADR=MEMPAG (A)          ;データ送信アドレスの設定
TRN_BNK=MEMBNK (A)          ;データ送信数1バイトの設定
COM_VOL=#1
SETI COMM_F
SETI TR_CG1F
break
case ENTER:
BACPAG=MEMPAG (A)           ;メモリ・アドレスのバックアップ
BACBNK=MEMBNK (A)
B=#ADC_DAT                   ;会社データの検索
C=#16                         ;比較アドレスの設定
DE=#LCDDS+150
H=#ADRSTA                     ;検索開始アドレスの設定
L=#0
M_BANK=#0
CALL !BANKSET
CALL !SAVADR                 ;会社データの検索
if_bit(!CMPERF)              ;会社データ一致あり
D=MEMPAG (A)                 ;一致処理
E=#16
HL=#LCDDS                     ;会社データをLCD表示エリアに設定
CALL !DT1PSET
else
CALL !LCD_DSC                 ;不一致処理(表示クリア)
HL=#LCDDS                     ;会社名を1行目に設定
D=BACPAG (A)
E=#16+150
M_BANK=BACBNK (A)
CALL !BANKSET
CALL !DT1LSET
endif
endif
MODE2=#3                     ;会社データ表示モードに設定
SETI MD_CG2F
CLR1 CR_DSPF
MEMPAG=BACPAG (A)           ;メモリ・アドレスの復帰
MEMBNK=BACBNK (A)
ends
ENDIF
endif
RET

;*****
; 名刺(個人)会社データ表示モード
;*****
AM_COM:
if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD)
case CLEAR:                   ;前モードに設定

```

```

                MODE2=#2
                SETI   MD_CG3F
        ends
    endif
    RET

;*****
; 名刺(個人)先頭文字選択モード
;*****
AM_DEL:
    if_bit(MD_CG1F)
        CLRI   MD_CG1F
        INV_DS=#0                ;反転表示行削除
        SETI   IV_CHGF
        HL=#LCDDS                ;メニュー表示データの設定
        DE=#NA_SEA
        CALL   !DTILSET          ;LCD 1行目のデータを設定

        DE=#CLR1D
        CALL   !DTILSET          ;LCD 2行目のデータを設定

        DE=#ATOM_D
        CALL   !DTILSET          ;LCD 3行目のデータを設定

        DE=#CLR1D
        CALL   !DTILSET          ;LCD 4行目のデータを設定

        DE=#NTOZ_D
        CALL   !DTILSET          ;LCD 5行目のデータを設定

        DE=#CLR1D
        CALL   !DTILSET          ;LCD 6行目のデータを設定

        DE=#NEXT_P
        CALL   !DTILSET          ;LCD 7行目のデータを設定

        DE=#CLR1D
        CALL   !DTILSET          ;LCD 8行目のデータを設定

        MODE2=#0
        MODE3=#0
        MODE4=#0

        SETI   CR_DSPF          ;カーソル表示の設定
        CUR_ADR=#62            ;カーソル・アドレスを'A'の位置に設定
        SETI   LCDCMGF

    endif
    switch(MODE2)
    case 0:
        CALL   !AM_DSEL          ;先頭文字選択モード
        break
    case 1:
        CALL   !AM_DTI           ;名前選択モード
        break
    case 2:
        CALL   !AM_DDEL          ;データ表示モード
    ends
    RET

;*****
; 名刺(個人)名前選択モード(削除)
;*****
AM_DTI:
    if_bit(MD_CG2F)
        CLRI   MD_CG2F
        CLRI   CR_DSPF
        CALL   !NAM_DSP
        FOR_BNK=#0
        FOR_ADR=#ADRSTA
        REV_BNK=#0
        REV_ADR=#ADRSTA
        MODE3=#0
        SETI   MD_CG3F
    endif
    IF_BIT(MD_CG3F)

```

```

CLR1 MD_CG3F
SWITCH(MODE3) ;検索方向の指定
CASE 0:
  FOR(j=#0;j<#7;j++) ;1 ページ表示数のカウント
    H=FOR_ADR (A)
    L=#00H
    M_BANK=FOR_BNK (A) ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    B=#ADM_DAT
    C=#16
    CALL !SEA_FOR
    if_bit(ADSEAF)
      D=H (A) ;転送元アドレスの設定
      E=#16
      PUSH DE ;アドレス退避
      switch(j) ;転送先アドレスの設定
      case 0: ;2 行目に設定
        CALL !NEX_ADR
        CALL !NAM_DSP
        HL=#LCDDS+30
        break
      case 1: ;3 行目に設定
        HL=#LCDDS+60
        break
      case 2: ;4 行目に設定
        HL=#LCDDS+90
        break
      case 3: ;5 行目に設定
        HL=#LCDDS+120
        break
      case 4: ;6 行目に設定
        HL=#LCDDS+150
        break
      case 5: ;7 行目に設定
        HL=#LCDDS+180
        break
      case 6: ;8 行目に設定
        HL=#LCDDS+210
      ends
      POP DE
      CALL !DT1LSET
      H=D (A)
      CALL !ADR_INC
      FOR_ADR=H (A)
      FOR_BNK=M_BANK (A)
    elseif_bit(ADENDF) ;検索データなし
      break
    endif
  NEXT
  break
CASE 1:
  FOR(j=#0;j<#7;j++) ;1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A) ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    B=#ADM_DAT
    C=#16
    CALL !SEA_REV
    if_bit(ADSEAF)
      D=H (A) ;転送元アドレスの設定
      E=#16
      PUSH DE ;アドレス退避
      switch(j) ;転送先アドレスの設定
      case 0: ;8 行目に設定
        CALL !BEF_ADR
        CALL !NAM_DSP
        HL=#LCDDS+210
        break
      case 1: ;7 行目に設定
        HL=#LCDDS+180
        break
      case 2: ;6 行目に設定
        HL=#LCDDS+150
        break

```

```

        case 3:                ; 5行目に設定
            HL=#LCDDS+120
            break
        case 4:                ; 4行目に設定
            HL=#LCDDS+90
            break
        case 5:                ; 3行目に設定
            HL=#LCDDS+60
            break
        case 6:                ; 2行目に設定
            HL=#LCDDS+30
        ends
        POP    DE
        CALL  !DT1LSET
        H=D (A)
        CALL  !ADR_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
    elseif_bit(ADENDF)        ; 検索データなし
        break
    endif
NEXT
ENDS
INV_DS=#2
SET1  IV_CHGF
ENDIF

IF_BIT(KY_SIGF)              ; キー入力あり
CLR1  KY_SIGF
switch(KEY_CD)
case CUR_UP:                  ; 反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SET1  IV_CHGF        ; 反転表示行変更状態に設定
    endif
    break
case CUR_DN:                  ; 反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)      ; 転送先アドレスの設定
        case 2:              ; 3行目に設定
            HL=#LCDDS+60
            break
        case 3:              ; 4行目に設定
            HL=#LCDDS+90
            break
        case 4:              ; 5行目に設定
            HL=#LCDDS+120
            break
        case 5:              ; 6行目に設定
            HL=#LCDDS+150
            break
        case 6:              ; 7行目に設定
            HL=#LCDDS+180
            break
        case 7:              ; 8行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!="#SPC) (A)   ; 次の行に名前あり
            INV_DS++
            SET1  IV_CHGF    ; 反転表示行変更状態に設定
        endif
    endif
    break
case CUR_RI:
    MODE3=#1
    SET1  MD_CG3F
    break
case CUR_LE:
    MODE3=#0
    SET1  MD_CG3F
    break
case CLEAR:
    MODE2=#0
    SET1  MD_CG1F
    break

```

```

case DELETE:
  BACPAG=MEMPAG (A)           ;メモリ・アドレスのバックアップ
  BACBNK=MEMBNK (A)
  B=#ADM_DAT                 ;名刺(個人)データの検索
  C=#16                      ;比較アドレスの設定
  switch(INV_DS)             ;転送先アドレスの設定
  case 2:                     ;2行目に設定
    DE=#LCDDS+30
    break
  case 3:                     ;3行目に設定
    DE=#LCDDS+60
    break
  case 4:                     ;4行目に設定
    DE=#LCDDS+90
    break
  case 5:                     ;5行目に設定
    DE=#LCDDS+120
    break
  case 6:                     ;6行目に設定
    DE=#LCDDS+150
    break
  case 7:                     ;7行目に設定
    DE=#LCDDS+180
    break
  case 8:                     ;8行目に設定
    DE=#LCDDS+210
  ends
  if([DE]!="#SPC) (A)
    H=#ADRSTA                 ;検索開始アドレスの設定
    L=#0
    M_BANK=#0
    CALL !BANKSET
    CALL !SAVADR              ;名刺(個人)データの検索
    CALL !DT_DEL              ;データ数1減算
    ADM_VOL--
  endif
  FOR_ADR=#ADRSTA
  FOR_BNK=#0
  SET1 MD_CG2F
  MODE3=#0
  SET1 MD_CG3F
  break
case ENTER:
  CLR1 ADSEAF
  M_BANK=#0                  ;外部メモリを開始位置に設定
  CALL !BANKSET
  H=#ADRSTA
  L=#0
  B=#ADM_DAT                 ;比較データを名刺(個人)データに設定
  C=#16                      ;1ライン比較の開始アドレスを設定(HL+C)
  switch(INV_DS)             ;検索対象行アドレスの設定
  case 2:                     ;2行目に設定
    DE=#LCDDS+30
    break
  case 3:                     ;3行目に設定
    DE=#LCDDS+60
    break
  case 4:                     ;4行目に設定
    DE=#LCDDS+90
    break
  case 5:                     ;5行目に設定
    DE=#LCDDS+120
    break
  case 6:                     ;6行目に設定
    DE=#LCDDS+150
    break
  case 7:                     ;7行目に設定
    DE=#LCDDS+180
    break
  case 8:                     ;8行目に設定
    DE=#LCDDS+210
  ends
  if([DE]!="#SPC) (A)       :
    repeat

```



```

CALL !SAVADR ;メモリ・アドレスを設定
if_bit(CY)
CALL !ADR_INC
endif
if([HL]==#VAC_DAT) (A) ;検索アドレスのデータは空きデータ
break ;登録データなし→検索終了
else
NODE2=#2 ;登録データあり→データ表示
SETI MD_CG3F
endif
until_bit(!CMPERF)

MEMBNK=M_BANK (A)
MENPAG=H (A)
endif
break
case UP:
if(SEACHR<#'Z') ;検索文字'A'-'Y'
SETI MD_CG2F
SEACHR++ ;次の文字に変更
endif
break
case DOWN:
if(SEACHR>#'A') ;検索文字'B'-'Z'
SETI MD_CG2F
SEACHR-- ;前の文字に変更
endif
ends
ENDIF
RET

;*****
; 名刺(個人)名前検索モード
;*****
AM_DDEL:
if_bit(MD_CG3F) ;修正データの変更
CLR1 MD_CG3F
M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
CALL !BANKSET
D=MENPAG (A)
E=#16
HL=#LCDDS ;LCD表示アドレスの設定
CALL !DTIPSET ;表示データの転送
CLR1 CR_DSPF ;カーソル表示なし
INV_DS=#0
SETI IV_CHGF
SETI LCDCMGF
endif
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD)
case CLEAR: ;前モードに設定
NODE2=#0
SETI MD_CG1F
break
case UP:
M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
CALL !BANKSET
H=MENPAG (A) ;表示データのページ設定
L=#0
if(H==#ADREND) (A) ;バンク最終アドレス
if(M_BANK<#3) ;バンク0-バンク2
M_BANK++ ;バンク更新
CALL !BANKSET
H=#ADRSTA
if([HL]==#ADM_DAT) (A) ;次データ=名刺(個人)データ
SETI MD_CG3F
MENPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
endif
else
H++ ;次データ参照
if([HL]==#ADM_DAT) (A)
SETI MD_CG3F ;アドレスの更新

```

```

MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
break
case DOWN:
M_BANK=MEMBNK (A)           ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG (A)                 ;表示データのページ設定
L=#0
if(H=#ADRSTA) (A)           ;バンク先頭アドレス
if(M_BANK>#0)               ;バンク0-バンク2
M_BANK--                    ;バンク更新
CALL !BANKSET
H=#ADREND
if([HL]==#ADM_DAT) (A)      ;次データ=名刺(個人)データ
SET1 MD_CG3F
MEMPAG=H (A)                ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H--                           ;次データ参照
if([HL]==#ADM_DAT) (A)      ;アドレスの更新
SET1 MD_CG3F
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
break
case DELETE:
H=MEMPAG (A)
L=#00H
M_BANK=MEMBNK (A)
CALL !BANKSET
CALL !DT_DEL
ADM_VOL--
MODE2=#0
SET1 MD_CG1F
ends
ENDIF
RET

;*****
; 名刺(個人)名前表示設定
;*****
NAM_DSP:
CALL !LCD_DSC
HL=#LCDDS
DE=#NAM_DAT                 ;1行目の表示データを設定
CALL !DT1LSET
!LCDDS+11=SEACHR (A)        ;検索文字の表示
SET1 LCDCHGF
RET
END

```

4.5 名刺（会社）モード

名刺（会社）モードでは、モード選択や、データの登録、修正、検索、削除を行っています。

4.5.1 登録データのフォーマット

名刺（会社）データの登録フォーマットを図4 - 7に示します。

図4 - 7 名刺（会社）データの登録フォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xx00H	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
xx10H	データの種類															
xx20H																
xx30H	LCD表示データ240バイト (30文字×8行)															
xx40H																
xx50H																
xx60H																
xx70H																
xx80H																
xx90H																
xxA0H																
xxB0H																
xxC0H																
xxD0H																
xxE0H																
xxF0H																

4.5.2 モード選択条件

名刺（会社）モードの遷移状態の条件を表4 - 6に示します。

表4 - 6 名刺（会社）モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
2	0	-	-	-	名刺（会社）モード選択
	1	0	-	-	名刺（会社）会社名登録モード
		1	-	-	名刺（会社）電話番号登録モード
		2	-	-	名刺（会社）FAX番号登録モード
		3	-	-	名刺（会社）住所登録モード
		4	-	-	名刺（会社）メモ内容登録モード
		5	-	-	メモリ・オーバフロー表示モード
	2	0	-	-	先頭文字選択モード
		1	-	-	会社名選択モード
		2	-	-	データ修正モード
	3	0	-	-	先頭文字選択モード
		1	-	-	会社名選択モード
		2	-	-	データ表示モード
		3	-	-	会社員一覧表示モード
	4	0	-	-	先頭文字選択モード
		1	-	-	会社名選択モード
		2	-	-	データ表示モード

備考 - : don't care

4.5.3 登録処理

登録モードでは、ENTキー入力によりMODE2の値を更新し、CLRキー入力のときは前モードに戻すためMODE1を0に設定します。登録データは、文字データのキー入力により、カーソルのあるアドレス(CUR_ADR)に文字データを書き込みます。すべてのデータ入力後(メモ入力後のENTキー)、次の順で登録処理を行います。

メモリ最終ページの先頭アドレスの確認をします。

バンク3(P61, P60 = 1, 1)のBF00H番地がFFH(空きエリア)以外のときはメモリ・オーバーフローにします(MODE2 = 6に設定)。

データを登録するアドレスを検索します。

検索条件: $\times \times 00H$ が03H以上、または、 $\times \times 00H$ が02Hで入力した名前のデータが、表示コードで比較して小さい値のとき

検索したアドレスにデータを登録します。表示データ(240文字)は、 $\times \times 10H$ - $\times \times FFH$ に登録し、 $\times \times 00H$ に02Hを、 $\times \times 01H$ - $\times \times 0FH$ に00Hに登録します。検索アドレス以降のデータは、次のアドレス(+100H)にシフトします。

データ数(ADC_VOL)を+1します。

4.5.4 データ選択処理

データ選択処理には、先頭文字選択モード(MODE2 = 0)と、会社名選択モード(MODE2 = 1)の2種類があります。

(1) 先頭文字選択モード

先頭文字選択モードでは、`,`、`.`、`;` キー入力により、カーソルのアドレス(CUR_ADR)の値に加算、減算をしてカーソルの位置を変更しています。ENTキー入力により、先頭文字を決定し(SEACHRにA-Zを設定)、会社名選択モードに設定します(MODE2 = 1)。ただし、NEXT PAGEを選択したときは、データ登録メモリの $\times \times 00H$ に02Hが格納されている先頭アドレスを検索し、次のモード(MODE2 = 2)に移行します。検索データがないときは、名刺(会社)モード選択に戻ります(MODE1 = 0)。

図4-8 名刺(会社)データの検索

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\times \times 00H$	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

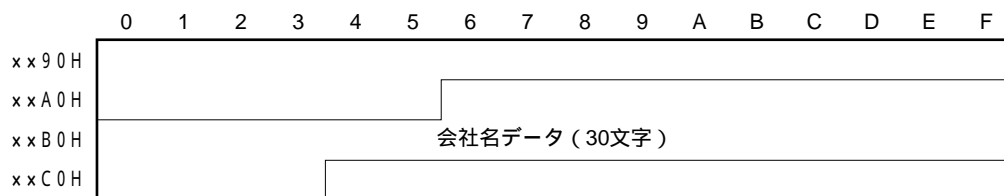
02Hで格納されている先頭を検索

4.5.6 検索処理

検索モードでは、ENTキー入力により、会社員一覧表示モードになります。会社員一覧表示モードでは、名刺（会社）の会社名のデータ（図4-9参照）と、名刺（個人）データ（ $\times \times 00H = 01H$ ）の $\times \times A6H$ から30文字のデータ（図4-10参照）が一致しているアドレスを検索し、一致したときの名前データをLCD表示エリアに格納し、会社員一覧モードに移行します（MODE2=3）。検索データは、1回の処理で7個（表示可能な範囲）まで表示します。表示している会社員の先頭/最終アドレスをメモリ（REV_ADR/FOR_ADR）に格納し、次の検索時の開始アドレスとして使用します。

検索データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの+100H/-100Hの $\times \times 00H$ データが02Hのときは、表示データの変更をします。

図4-10 名刺（個人）の会社名データ



4.5.7 削除処理

削除モードでは、DELキー入力により表示中のデータを削除します。データ削除後は、個人（名刺）モード選択に戻ります（MODE1=0）。CLRキー入力の際は、前モードに戻すためMODE1を0に設定します。

削除データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの+100H/-100Hの $\times \times 00H$ データが01Hのときは、表示データの変更をします。次の順でデータ削除を行っています。

削除データのアドレスを検索します。

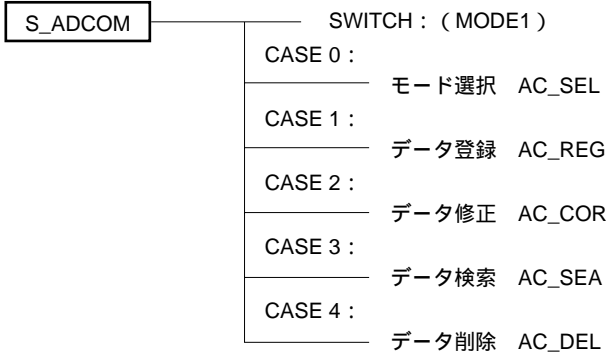
検索したアドレスを、次のアドレス（+100H）データから最終アドレス（P61,P60=1,1,BFFFH）までシフトし、上書きすることでデータを削除します。

データ数（ADC_VOL）を-1します。

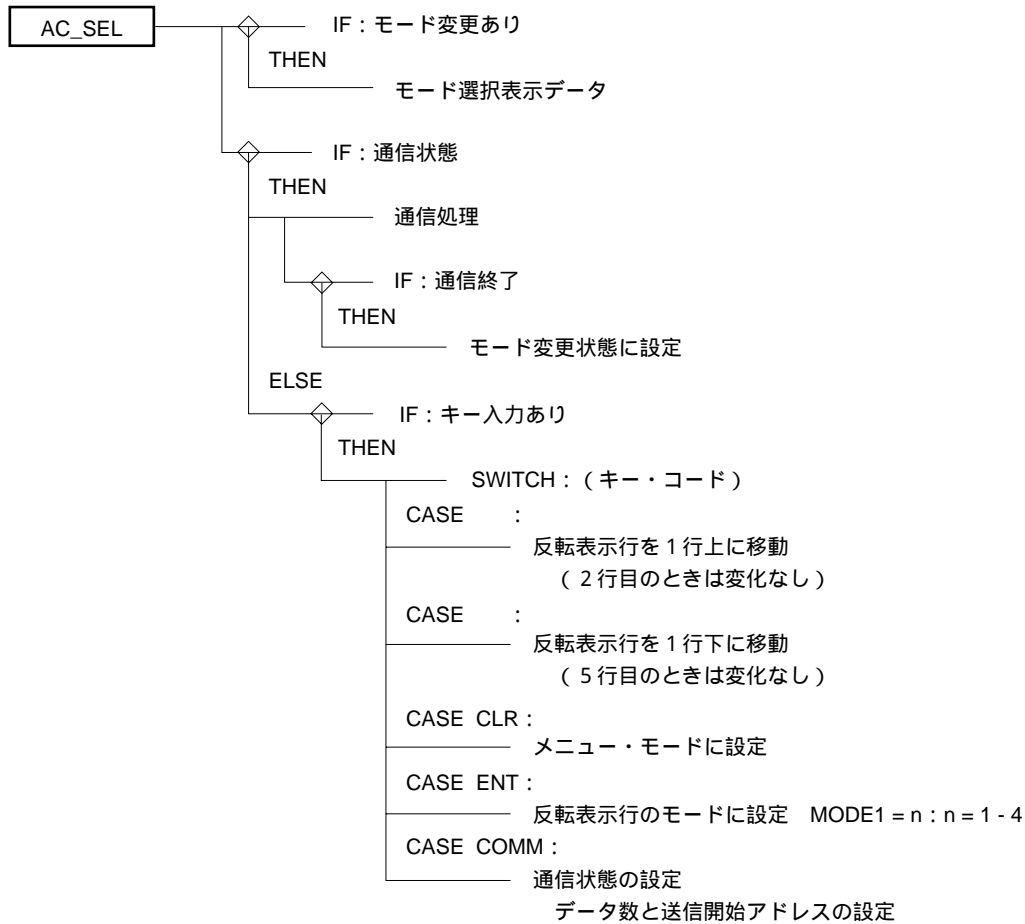
最終アドレス（BF00H-BFFFH）に空きデータ（FFH）を書き込みます。

4.5.8 SPDチャート

名刺(会社)モード

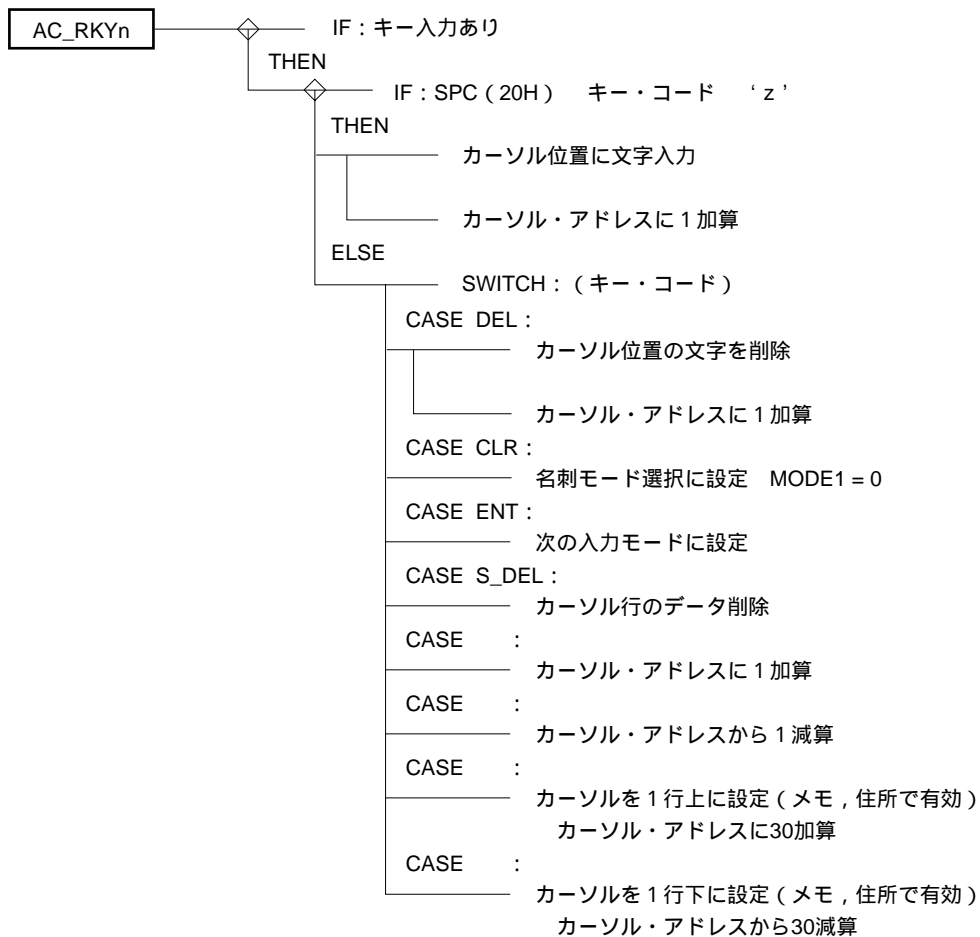
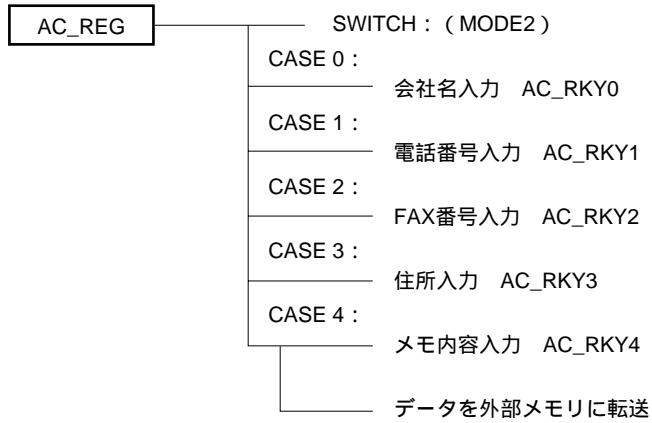


・モード選択



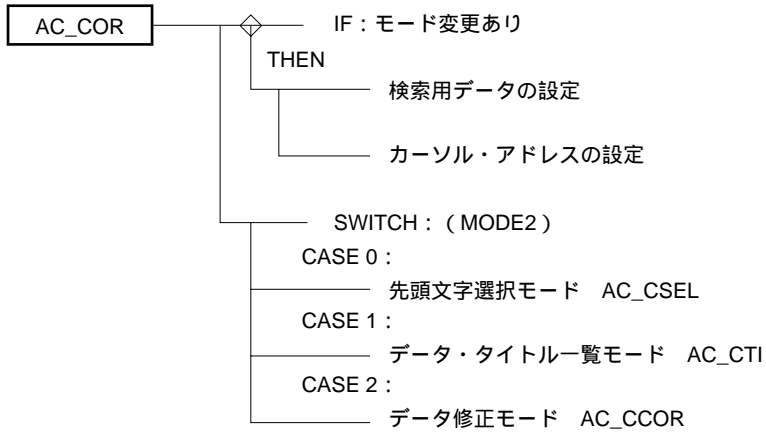
名刺（会社）モード

・登録モード



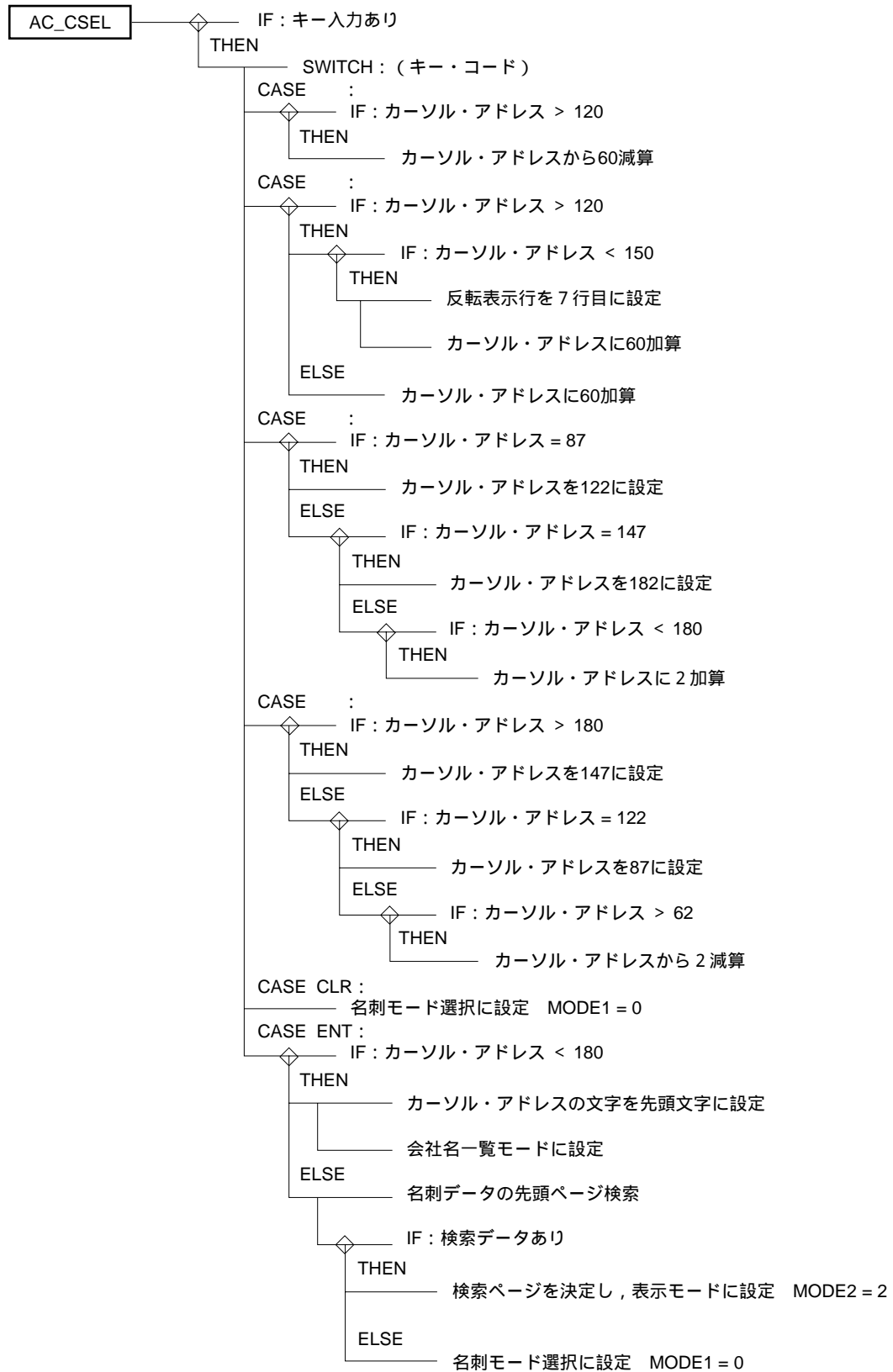
名刺（会社）モード

・修正モード



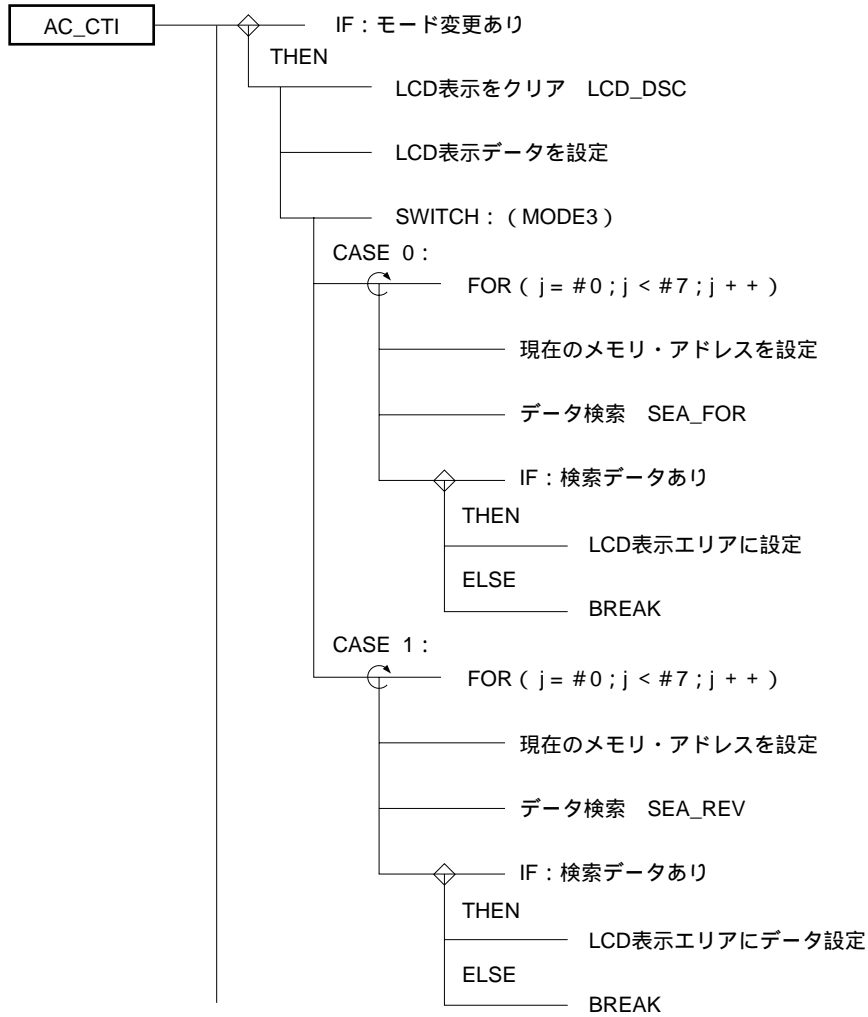
名刺（会社）モード

・修正モード



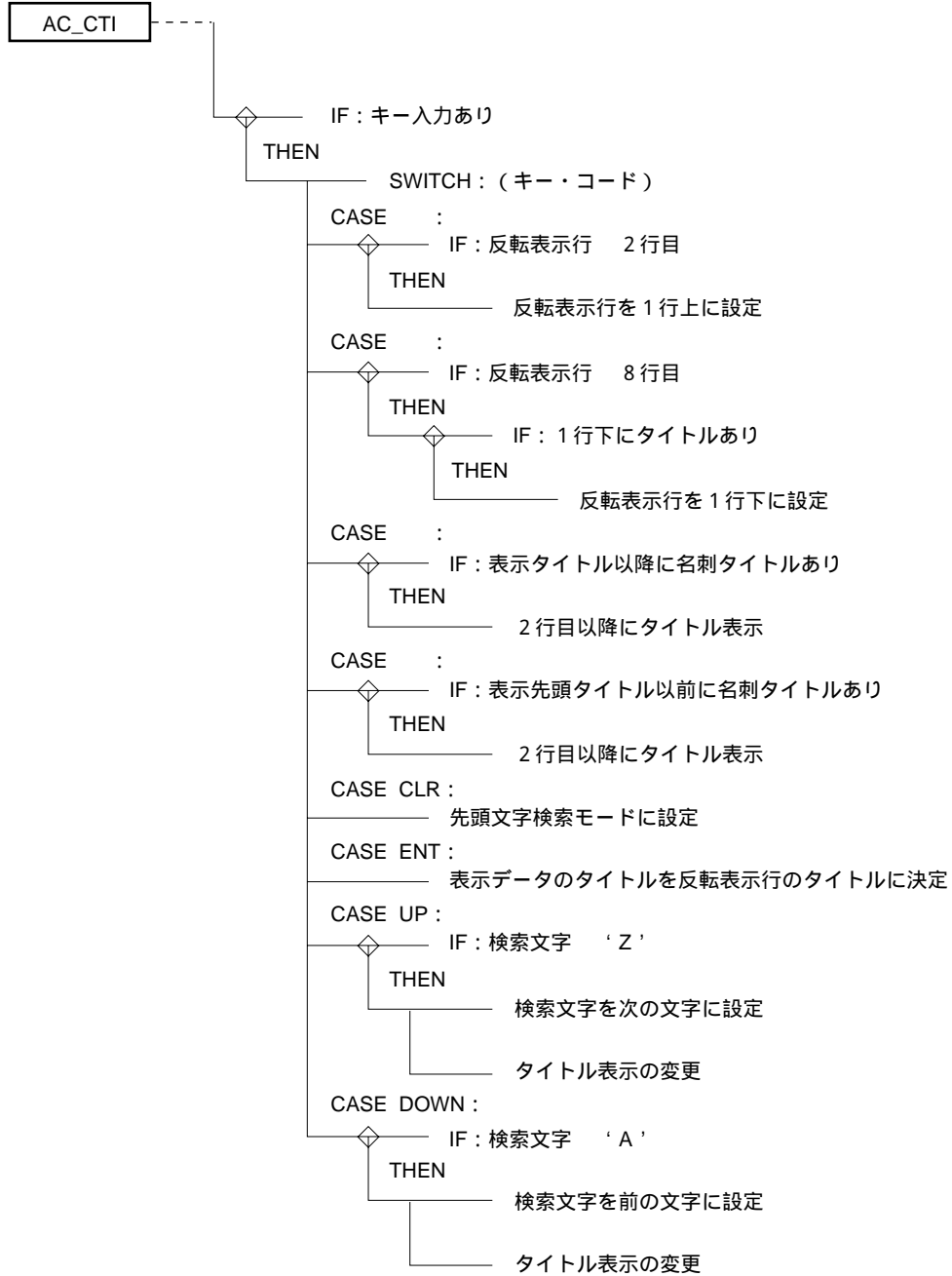
名刺(会社)モード

・修正モード



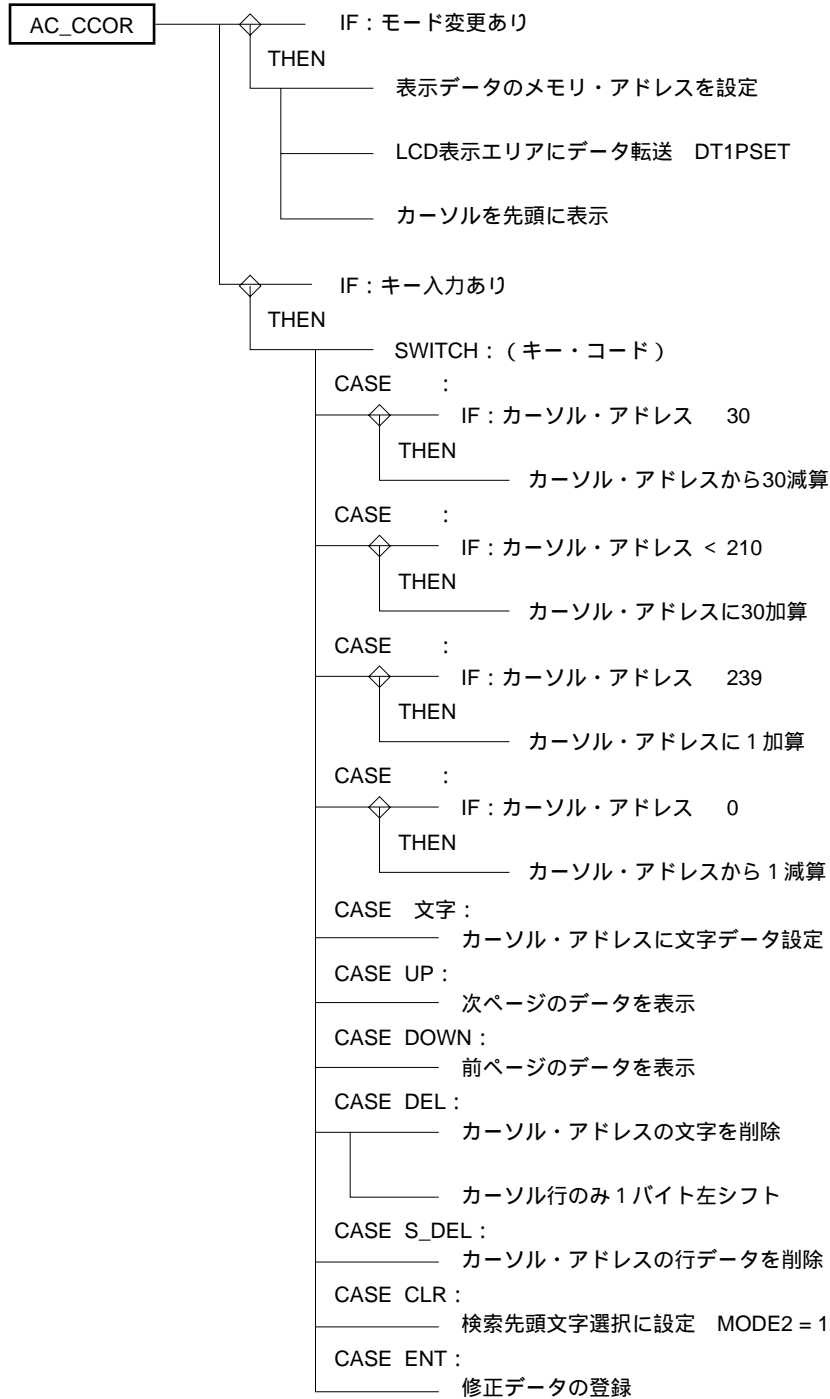
名刺(会社)モード

・修正モード



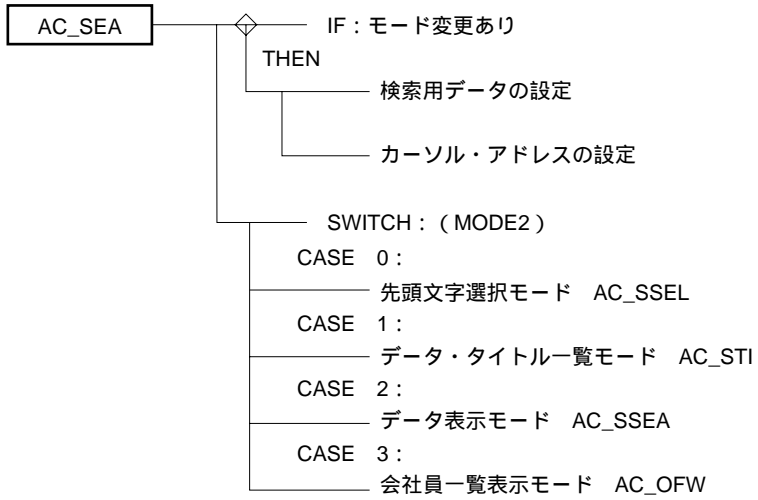
名刺（会社）モード

・修正モード



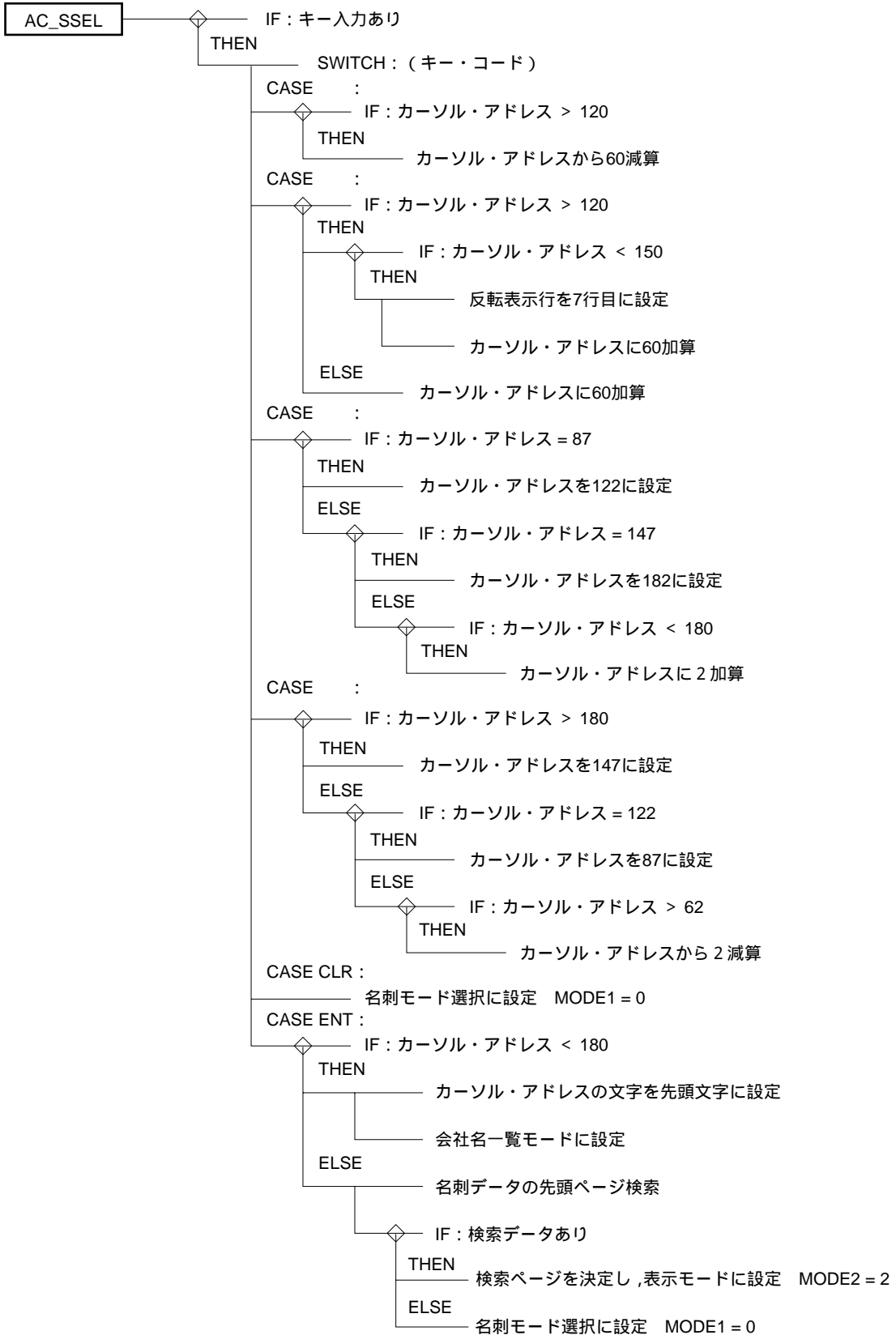
名刺(会社)モード

・検索モード



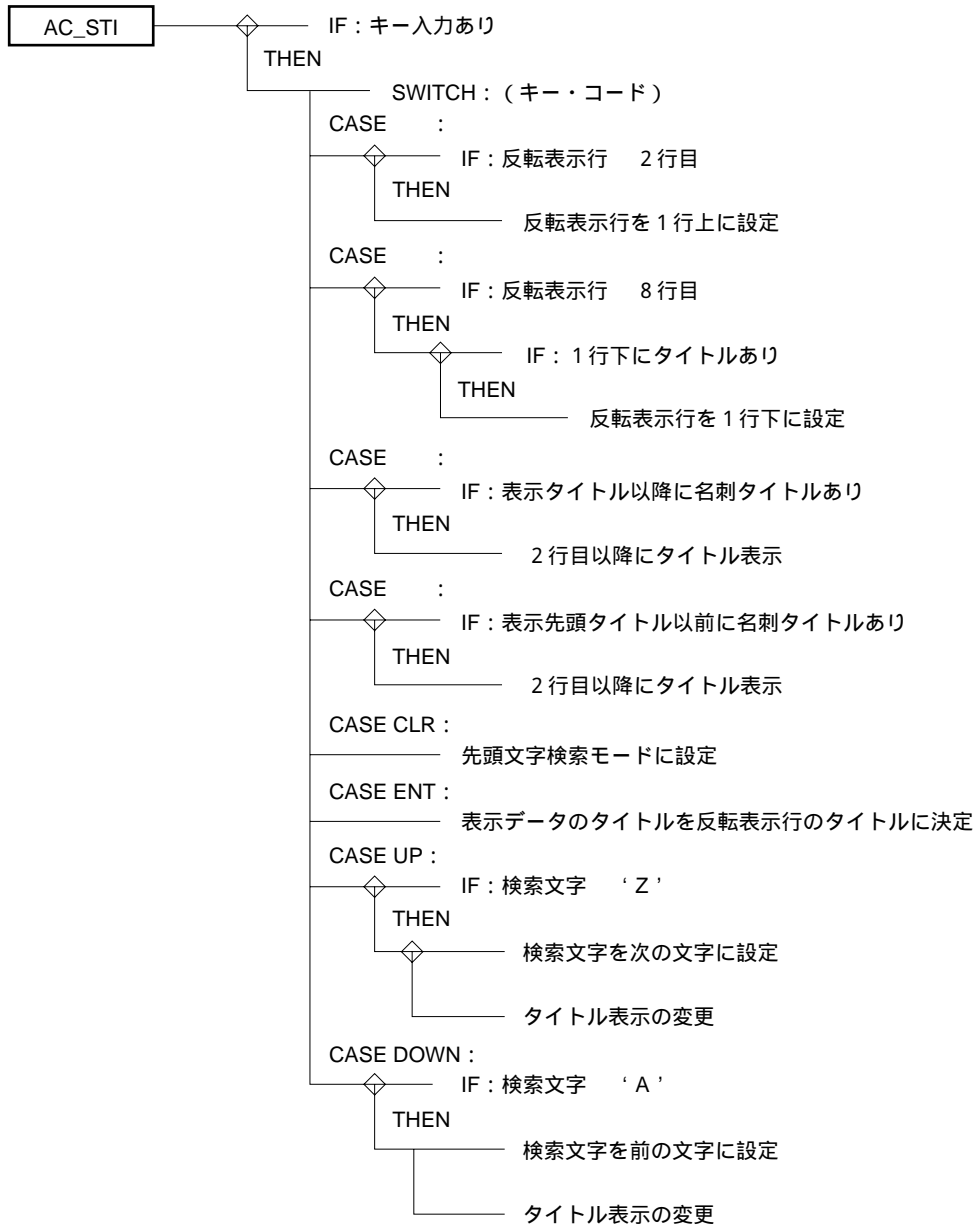
名刺（会社）モード

・検索モード



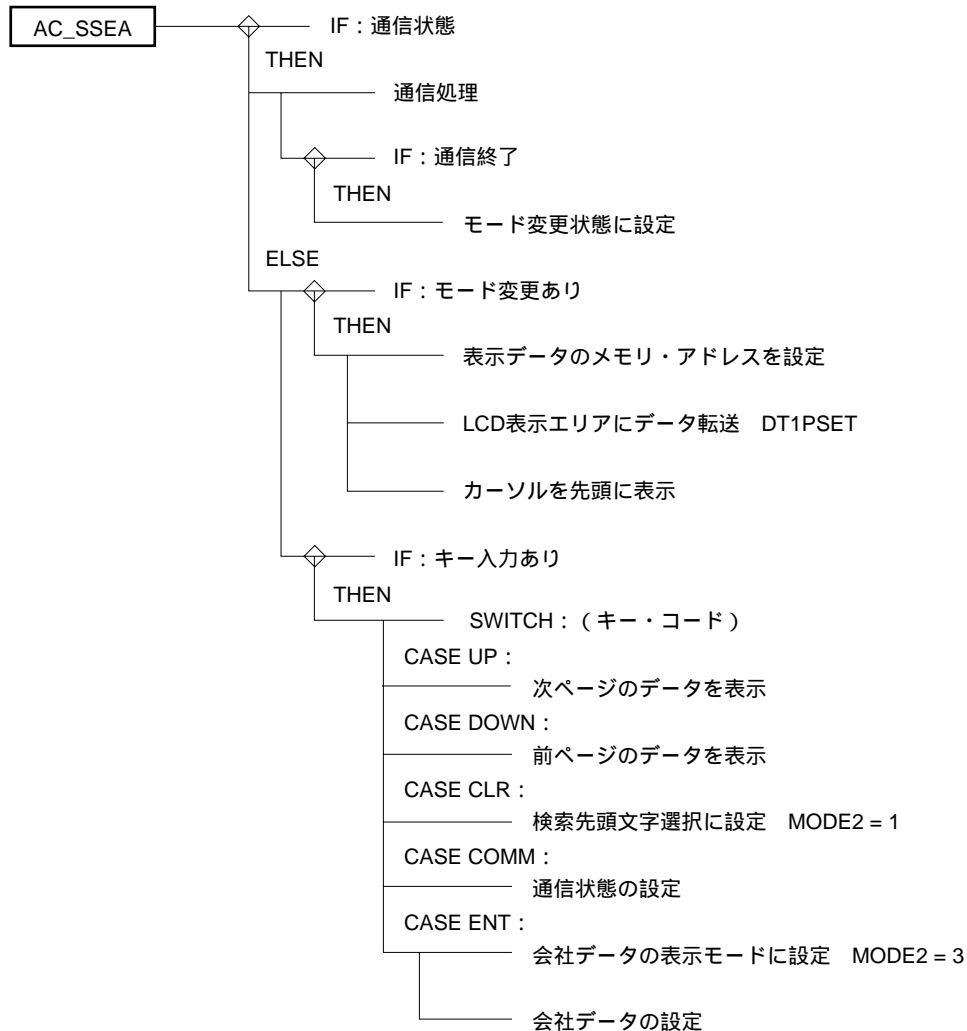
名刺(会社)モード

・検索モード



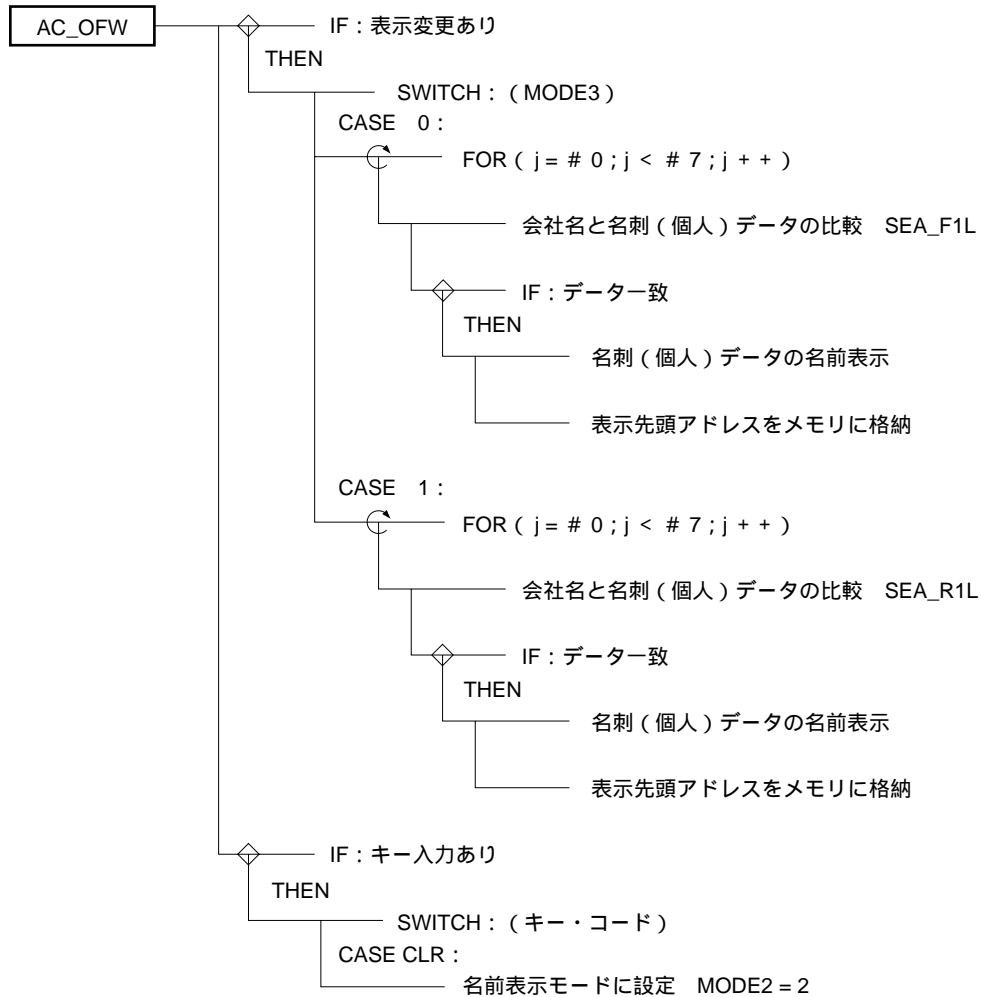
名刺（会社）モード

・検索モード



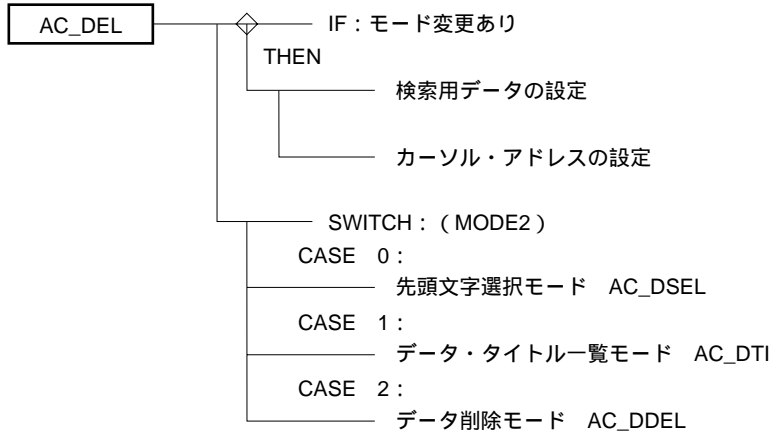
名刺(会社)モード

・検索モード：会社員一覧



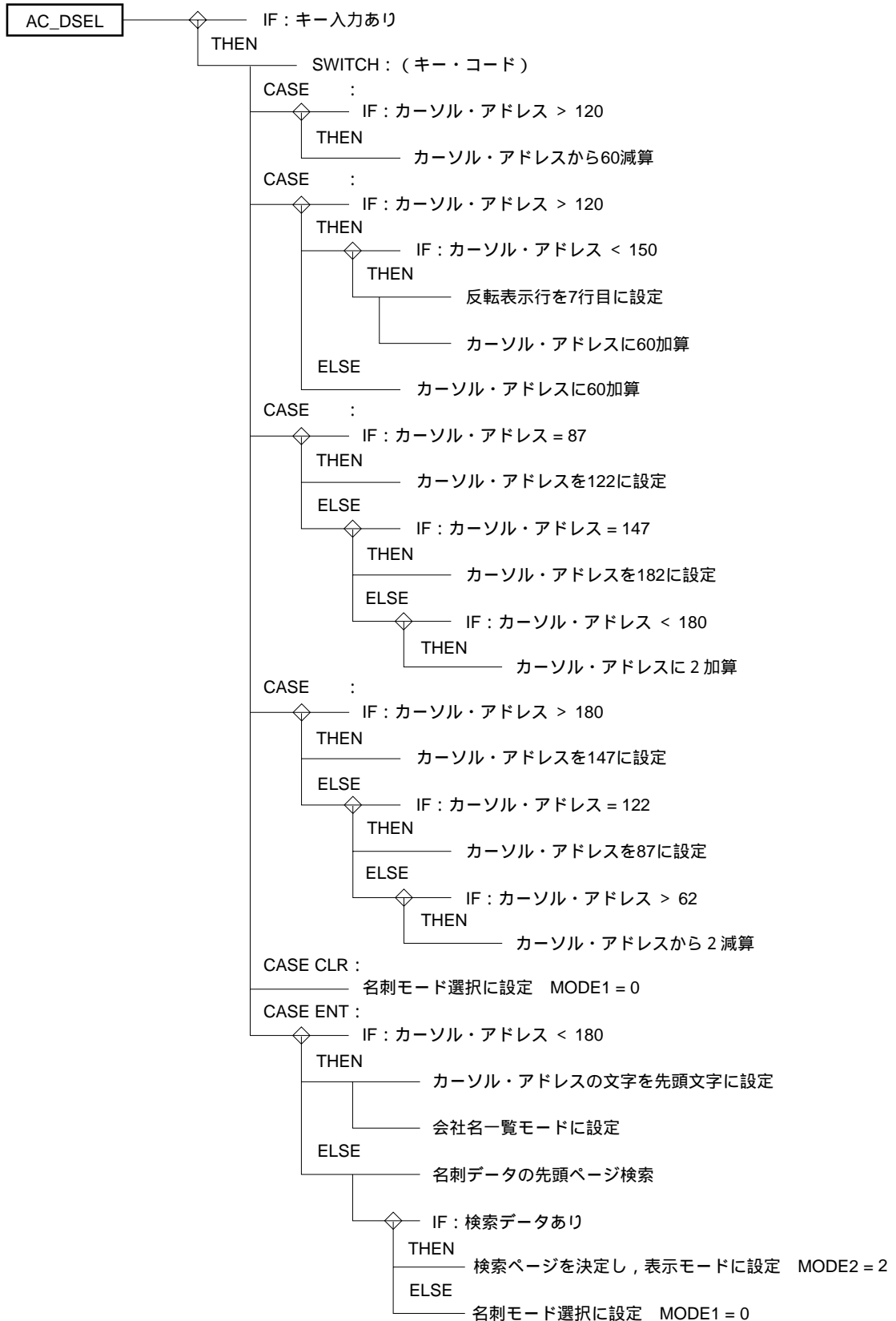
名刺（会社）モード

・削除モード



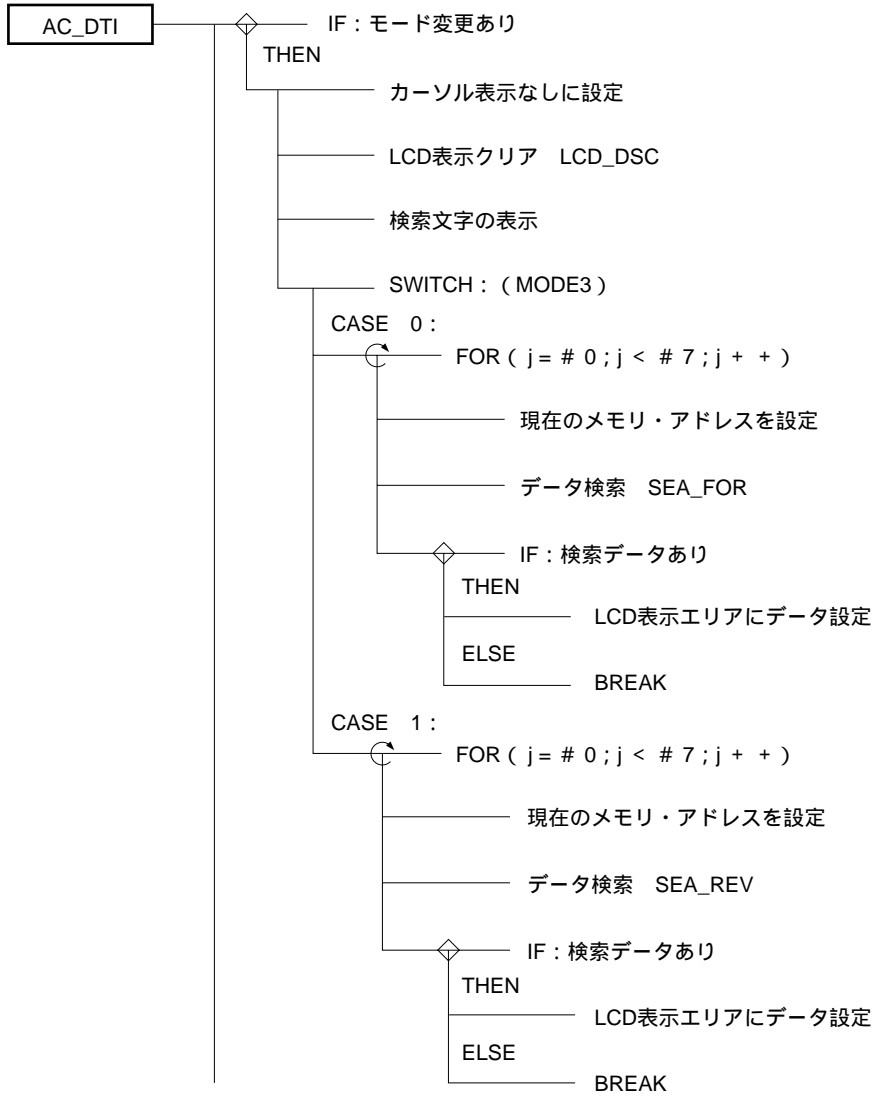
名刺（会社）モード

・削除モード



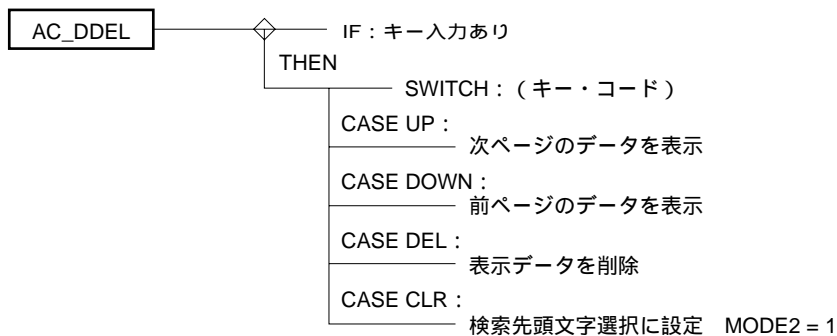
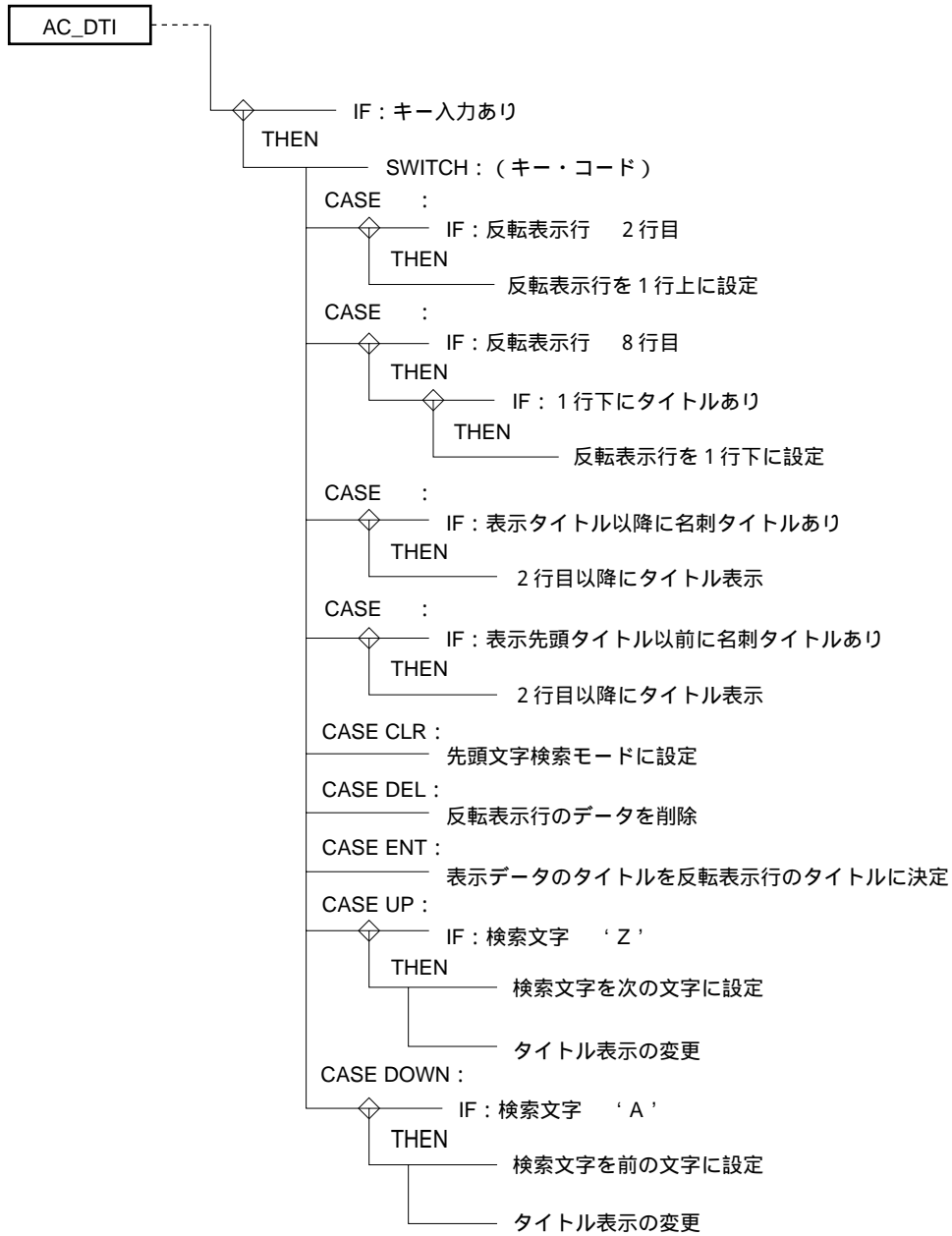
名刺(会社)モード

・削除モード



名刺(会社)モード

・削除モード



4.5.9 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYVK, INV_DS
EXTRN  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, NEMPAG, MEMBNK, BACPAG, BACBNK
EXTRN  FOR_ADR, FOR_BNK, REV_ADR, REV_BNK, SEACHR, i, j
EXTRN  COM_VOL, TRN_ADR, TRN_BNK, VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DT1LSET, DT1PSET, LCD_C, LCD_D, DT_SAV, STA_SEA, S_COMM
EXTRN  SAVADR, DTCOMP, DT_DEL, BANKSET, SEA_FOR, SEA_REV, SEA_FIL, SEA_RIL, ADR_INC, ADR_DEC, NEX_ADR, BEF_ADR
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, IV_CHGF, COMM_F, DS_BRAF
EXTBIT  CR_DSPF, ADENDF, ADSEAF, CMPEF, TR_CG1F, TR_CG2F, TR_CG3F, LCDCHGF

PUBLIC  S_ADCOM, ADC_VOL

$      INCLUDE (LCDCMD.INC)
$      INCLUDE (LCDT_EX.INC)

$LIST

      DSEG      SADDR
ADC_VOL:DS      1                                ;会社データ数

;*****
;      名刺(会社)分岐処理
;*****
ADCSEG      CSEG
S_ADCOM:
      switch(MODE1)
      case 0:
            CALL      IAC_SEL                        ;モード選択
            break
      case 1:
            CALL      IAC_REG                        ;データ登録
            break
      case 2:
            CALL      IAC_COR                        ;データ修正
            break
      case 3:
            CALL      IAC_SEA                        ;データ検索
            break
      case 4:
            CALL      IAC_DEL                        ;データ削除
      ends
      RET

;*****
;      名刺(会社)モード選択処理
;*****
AC_SEL:
      if_bit(MD_CG1F)
            CLR1      MD_CG1F
            HL=#LCDDS                                ;メニュー表示データの設定
            DE=#ADC_D                                ;LCD 1行目のデータを設定
            CALL      IDTILSET

            DE=#REG_D                                ;LCD 2行目のデータを設定
            CALL      IDTILSET

            DE=#COR_D                                ;LCD 3行目のデータを設定
            CALL      IDTILSET

            DE=#SEA_D                                ;LCD 4行目のデータを設定
            CALL      IDTILSET

            DE=#DEL_D                                ;LCD 5行目のデータを設定
            CALL      IDTILSET

            DE=#CLR1D                                ;LCD 6行目のデータを設定

```



```

CALL    !DTILSET

DE=#CLR1LD                ;LCD 7行目のデータを設定
CALL    !DTILSET

DE=#CLR1LD                ;LCD 8行目のデータを設定
CALL    !DTILSET

MODE2=#0
MODE3=#0
MODE4=#0
SET1    MD_CG2F
SET1    MD_CG3F
SET1    MD_CG4F
CLR1    CR_DSPF
SET1    IV_CHGF                ;反転表示行を2行目に設定
INV_DS=#2
SET1    LCDCHGF
endif

if_bit(COMM_F)            ;通信状態
CALL    !S_COMM            ;通信処理
if_bit(!COMM_F)          ;通信終了
SET1    MD_CG1F            ;通信前の状態に設定
endif
else
if_bit(KY_SIGF)
CLR1    KY_SIGF
switch(KEY_CD)            ;キー・コード分岐
case CUR_UP:
if(INV_DS>#2)
INV_DS--
SET1    IV_CHGF            ;反転表示行変更状態に設定
endif
break
case CUR_DN:                ;反転表示行を下に移動
if(INV_DS<#5)
INV_DS++
SET1    IV_CHGF            ;反転表示行変更状態に設定
endif
break
case CLEAR:
MODE0=#0                ;前画面モードに変更
SET1    MD_CG0F
break
case ENTER:                ;反転表示行のモード実行
A=INV_DS
A--
MODE1=A
SET1    MD_CG1F            ;モード変更状態に設定
break
case COMM:                ;通信状態に設定
COM_VOL=ADC_VOL (A)    ;送信データ数の設定
B=#ADC_DAT              ;検索データの設定
CALL    !STA_SEA
TRN_ADR=H (A)            ;送信データ数の設定
TRN_BNK=M_BANK (A)
SET1    COMM_F
SET1    TR_CG1F
ends
endif
endif
RET

;*****
;  名刺(会社)データ登録処理
;*****
AC_REG:
switch(MODE2)
case 0:
if_bit(MD_CG2F)
CLR1    MD_CG2F
INV_DS=#0                ;反転表示行削除
SET1    IV_CHGF

```

```

CALL !LCD_DSC
HL=#LCDDS
DE=#COM_D
CALL !DTILSET
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AC_RKY0
break
case 1:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+30
DE=#TEL_D
CALL !DTILSET
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AC_RKY1
break
case 2:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+60
DE=#FAX_D
CALL !DTILSET
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AC_RKY2
break
case 3:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+90
DE=#ADR_D
CALL !DTILSET
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AC_RKY3
break
case 4:
if_bit(MD_CG2F)
CLR1 MD_CG2F
HL=#LCDDS+150
DE=#MEMO_D
CALL !DTILSET
SETI DS_ERAF
SETI LCDCHGF
endif
CALL !AC_RKY4
break
case 5:
if_bit(MD_CG2F)
CLR1 MD_CG2F
CALL !LCD_DSC
HL=#LCDDS+90
DE=#MEM_OV
CALL !DTILSET
CLR1 CR_DSPF
SETI LCDCHGF
endif
CALL !AC_RKY5
ends
RET

;*****
; 名刺(会社)データ登録キー入力処理
;*****
AC_RKY0:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
CLR1 DS_ERAF

```

:カーソル・アドレスの設定
:0000H(1行目)に設定
:カーソル・アドレスの設定
:0000H+30(2行目)に設定
:カーソル・アドレスの設定
:0000H+60(3行目)に設定
:カーソル・アドレスの設定
:0000H+90(4行目)に設定
:カーソル・アドレスの設定
:0000H+150(6行目)に設定
:カーソル・アドレスの設定
;エラー表示
:0000H+90(4行目)に設定
:キー入力あり
:1ライン・データ削除

```

HL=#LCDDS ;カーソル・アドレスの設定
DE=#CLRILD
CALL !DTILSET ;0 0 0 0 H (1行目)に設定
CUR_ADR=#0
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
D=#30
CALL !AC_RDEL
break
case CLEAR: ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER: ;電話入力モードに設定
MODE2=#1
SET1 MD_CG2F
CLR1 CR_DSPF
break
case S_DEL: ;1ライン・データ削除
HL=#LCDDS ;カーソル・アドレスの設定
DE=#CLRILD
CALL !DTILSET
CUR_ADR=#0 ;カーソル・アドレスを先頭に設定
break
case CUR_RI:
if(CUR_ADR<#29) ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
break
case CUR_LE:
if(CUR_ADR>#0) ;カーソル・アドレス1行目先頭アドレス以上
CUR_ADR--
endif
break
ends
ELSE
B=CUR_ADR(A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD(A) ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#29) ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
ENDIF
ENDIF
RET

AC_RKY1:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
CLR1 DS_ERAF
HL=#LCDDS+30 ;カーソル・アドレスの設定
DE=#CLRILD
CALL !DTILSET ;0 0 0 0 H + 3 0 (2行目)に設定
CUR_ADR=#30
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
D=#60
CALL !AC_RDEL
break
case CLEAR: ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER: ;FAX入力モードに設定
MODE2=#2
SET1 MD_CG2F
CLR1 CR_DSPF
break
case S_DEL: ;1ライン・データ削除

```

```

HL=#LCDDS+30          ;カーソル・アドレスの設定
DE=#CLRILD
CALL !DTILSET
CUR_ADR=#30          ;カーソル・アドレスを先頭に設定
break
case CUR_RI:
  if(CUR_ADR<#59)    ;カーソル・アドレス 2 行目最終行以内
    CUR_ADR++
  endif
  break
case CUR_LE:
  if(CUR_ADR>#30)    ;カーソル・アドレス 2 行目先頭アドレス以上
    CUR_ADR--
  endif
ends
ELSE
B=CUR_ADR (A)        ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)   ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#59)    ;カーソル・アドレス 2 行目最終行以内
  CUR_ADR++
endif
ENDIF
ENDIF
RET

AC_RKY2:
IF_BIT(KY_SIGF)
CLR! KY_SIGF
if_bit(DS_ERAF)
CLR! DS_ERAF
HL=#LCDDS+60        ;カーソル・アドレスの設定
DE=#CLRILD
CALL !DTILSET      ;0000H+60(3行目)に設定
CUR_ADR=#60
SET! CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
  D=#90
  CALL !AC_RDEL
  break
case CLEAR:        ;前モードに設定
  NODE1=#0
  SET! MD_CG1F
  break
case ENTER:        ;住所入力モードに設定
  NODE2=#3
  SET! MD_CG2F
  CLR! CR_DSPF
  break
case S_DEL:        ;1ライン・データ削除
  HL=#LCDDS+60    ;カーソル・アドレスの設定
  DE=#CLRILD
  CALL !DTILSET
  CUR_ADR=#60    ;カーソル・アドレスを先頭に設定
  break
case CUR_RI:
  if(CUR_ADR<#89) ;カーソル・アドレス 3 行目最終行以内
    CUR_ADR++
  endif
  break
case CUR_LE:
  if(CUR_ADR>#60) ;カーソル・アドレス 3 行目先頭アドレス以上
    CUR_ADR--
  endif
ends
ELSE
B=CUR_ADR (A)        ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)   ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#89)    ;カーソル・アドレス 3 行目最終行以内
  CUR_ADR++
endif
endif

```

```

ENDIF
ENDIF
RET

AC_RKY3:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_BRAF)
CLR1 DS_ERAF
HL=#LCDDS+90 ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET ;0000H+90(4行目)に設定
CUR_ADR=#90
SET1 CR_DSPF
endif
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
D=#150
CALL !AC_RDEL
break
case CLEAR: ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER: ;会社入力モードに設定
MODE2=#4
SET1 MD_CG2F
CLR1 CR_DSPF
break
case S_DEL: ;1ライン・データ削除
if(CUR_ADR<#120)
HL=#LCDDS+90 ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET
CUR_ADR=#90 ;カーソル・アドレスを先頭に設定
else
HL=#LCDDS+120 ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET
CUR_ADR=#120 ;カーソル・アドレスを先頭に設定
endif
break
case CUR_RI:
if(CUR_ADR<#149) ;カーソル・アドレス5行目最終行以内
CUR_ADR++
endif
break
case CUR_LE:
if(CUR_ADR>#90) ;カーソル・アドレス4行目先頭アドレス以上
CUR_ADR--
endif
break
case CUR_UP:
if(CUR_ADR=#120)
CUR_ADR-=#30 ;カーソル・アドレスを1行上(+30)に設定
endif
break
case CUR_DN:
if(CUR_ADR<#120)
CUR_ADR+=#30 ;カーソル・アドレスを1行下(-30)に設定
endif
ends
ELSE
B=CUR_ADR(A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD(A) ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#149) ;カーソル・アドレス5行目最終行以内
CUR_ADR++
endif
ENDIF
ENDIF
RET

AC_RKY4:

```

```

IF_BIT(KY_SIGF)
  CLR1  KY_SIGF
  if_bit(DS_ERAF)
    CLR1  DS_ERAF
    HL=#LCDDS+150      ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL  !DT1LSET      ;0 0 0 0 H + 1 5 0 (6行目) に設定
    CUR_ADR=#150
    SET1  CR_DSPF
  endif
  IF(KEY_CD<#SPC || KEY_CD<#'z') ;文字コード以外?
    switch(KEY_CD)
      case DELETE:
        D=#240
        CALL  !AC_RDEL
        break
      case CLEAR: ;前モードに設定
        MODE1=#0
        SET1  MD_CG1F
        break
      case ENTER: ;データを外部メモリに登録
        M_BANK=#3
        CALL  !BANKSET
        A=10BF00H
        if(A==#VAC_DAT) ;最終アドレスのメモリ空きあり
          CLR1  ADSEAF
          M_BANK=#0 ;外部メモリを開始位置に設定
          CALL  !BANKSET
          H=#ADRSTA
          L=#0
          B=#ADC_DAT ;比較データを名刺(会社)データに設定
          C=#16 ;1ライン比較の開始アドレスを設定(HL+C)
          DE=#LCDDS ;比較先頭アドレス
          CALL  !SAVADR ;登録メモリ・アドレスを設定
          DE=#DTWORK ;転送用データの作成
          [DE]=#ADC_DAT(A) ;データ種類の設定
          DE++
          A=#0
          for(i=#1;i<#16;i++) ;2-15バイトのデータをクリア
            [DE]=A
            DE++
          next
          HL=#LCDDS ;LCD表示データをワーク・エリアに設定
          for(i=#0;i<#240;i++)
            [DE]=[HL](A)
            DE++
            HL++
          next
          DE=#DTWORK
          H=#MEMPAG(A)
          L=#0
          CALL  !DT_SAV ;データ登録
          ADC_VOL++
          MODE1=#0 ;登録終了
          SET1  MD_CG1F
        else
          SET1  MD_CG2F ;メモリ・アドレスに空きなし
          MODE2=#5 ;オーバフロー・モードに設定
        endif
        CLR1  CR_DSPF
        break
      case S_DEL: ;1ライン・データ削除
        if(CUR_ADR<#180)
          HL=#LCDDS+150 ;カーソル・アドレスの設定
          DE=#CLR1LD
          CALL  !DT1LSET
          CUR_ADR=#150 ;カーソル・アドレスを先頭に設定
        elseif(CUR_ADR<#210)
          HL=#LCDDS+180 ;カーソル・アドレスの設定
          DE=#CLR1LD
          CALL  !DT1LSET
          CUR_ADR=#180 ;カーソル・アドレスを先頭に設定
        else
          HL=#LCDDS+210 ;カーソル・アドレスの設定
          DE=#CLR1LD
        endif
    endswitch
  endif

```

```

        CALL      !DTILSET
        CUR_ADR=#210      ;カーソル・アドレスを先頭に設定
    endif
    break
case CUR_RI:
    if(CUR_ADR<#239)      ;カーソル・アドレス8行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#150)      ;カーソル・アドレス6行目先頭アドレス以上
        CUR_ADR--
    endif
    break
case CUR_UP:
    if(CUR_ADR<=#180)
        CUR_ADR-=#30      ;カーソル・アドレスを1行上(+30)に設定
    endif
    break
case CUR_DN:
    if(CUR_ADR<#210)
        CUR_ADR+=#30      ;カーソル・アドレスを1行下(-30)に設定
    endif
ends
ELSE
    B=CUR_ADR (A)          ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)      ;カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#239)
        CUR_ADR++
        endif
    ENDIF
ENDIF
ENDIF
RET

AC_RKY5:
    if_bit(KY_SIGF)
        CLR1  KY_SIGF
        switch(KEY_CD)
        case CLEAR:
            MODE1=#0      ;前モードに設定
            SET1  MD_CGIF
        ends
    endif
    RET

AC_RDEL:
    B=CUR_ADR (A)          ;現在のカーソル・アドレスを設定
    C=CUR_ADR (A)          ;データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(C<D) (A)
        [HL+B]=[HL+C] (A) ;データ転送
        B++
        C++
    endw
    [HL+B]=#SPC (A)        ;行最終アドレスにスペース・データを設定
    RET

;*****
;  名刺(会社)先頭文字選択モード
;*****
AC_COR:
    if_bit(MD_CGIF)
        CLR1  MD_CGIF
        INV_DS=#0      ;反転表示削除
        SET1  IV_CHGF
        HL=#LCDDS      ;メニュー表示データの設定
        DE=#NA_SEA
        CALL  !DTILSET      ;LCD1行目のデータを設定

        DE=#CLR1LD
        CALL  !DTILSET      ;LCD2行目のデータを設定

        DE=#ATOM_D

```

```

CALL !DTI!SET ;LCD 3行目のデータを設定

DE=#CLR!LD
CALL !DTI!SET ;LCD 4行目のデータを設定

DE=#NTOZ_D
CALL !DTI!SET ;LCD 5行目のデータを設定

DE=#CLR!LD
CALL !DTI!SET ;LCD 6行目のデータを設定

DE=#NEXT_P
CALL !DTI!SET ;LCD 7行目のデータを設定

DE=#CLR!LD
CALL !DTI!SET ;LCD 8行目のデータを設定

MODE2=#0
MODE3=#0
MODE4=#0

SETI CR_DSPF ;カーソル表示の設定
CUR_ADR=#62 ;カーソル・アドレスを'A'の位置に設定
SETI LCDCRGF

endif
switch(MODE2)
case 0:
CALL !AC_CSEL ;先頭文字選択モード
break
case 1:
CALL !AC_CTI ;名前選択モード
break
case 2:
CALL !AC_CCOR ;データ修正モード
ends
RET

;*****
; 名称(会社)先頭文字選択モード
;*****
AC_CSEL:
AC_SSEL:
AC_DSEL:
IF_BIT(KY_SIGF) ;キー入力あり
CLR! KY_SIGF
switch(KEY_CD)
case CUR_RI:
if(CUR_ADR==#86) ;カーソル・アドレス=M
CUR_ADR=#122 ;カーソル・アドレス=N
elseif(CUR_ADR==#146) ;カーソル・アドレス=Z
CUR_ADR=#182 ;7行目反転表示
INV_DS=#7
SETI IV_CHGF
elseif(CUR_ADR<#180) ;カーソル・アドレス上記以外の値
CUR_ADR+=#2 ;カーソル・アドレスを次の文字に設定
endif
break
case CUR_LE:
if(CUR_ADR>#180) ;反転表示行
CUR_ADR=#146 ;カーソル・アドレス=Z
INV_DS=#0
SETI IV_CHGF ;反転表示消去
elseif(CUR_ADR==#122) ;カーソル・アドレス=N
CUR_ADR=#86 ;カーソル・アドレス=M
elseif(CUR_ADR>#82) ;カーソル・アドレス上記以外の値
CUR_ADR-=#2 ;カーソル・アドレスを前の文字に設定
endif
break
case CUR_UP:
if(CUR_ADR>#120) ;カーソル・アドレスN以降
CUR_ADR-=#80 ;カーソル・アドレスを1行上(-80)に設定
INV_DS=#0
SETI IV_CHGF ;反転表示消去
endif
break

```



```

case CUR_DN:
  if(CUR_ADR>#120)                ;カーソル・アドレスN以降
    if(CUR_ADR<#150)              ;反転表示でない
      INV_DS=#7
      SET1  IV_CHGF                ;7行目反転表示
      CUR_ADR+=#60                ;カーソル・アドレスを1行下(+60)に設定
    endif
  else
    CUR_ADR+=#60                  ;カーソル・アドレスを1行下(+60)に設定
  endif
  break
case CLEAR:                        ;前モードに設定
  MODE1=#0
  SET1  MD_CG1F
  break
case ENTER:
  if(CUR_ADR<#180)                ;カーソル・アドレス=A-Z
    HL=#LCDDS
    B=CUR_ADR (A)
    SEACHR=[HL+B] (A)             ;検索文字を登録
    SET1  MD_CG2F
    SET1  MD_CG3F
    MODE2=#1                      ;名前選択モードに設定
    MODE3=#0
    BACPAG=MEMPAG (A)
    BACBANK=MEMBANK (A)
    CALL  !LCD_DSC                ;LCD表示エリアをクリア
  else
    B=#ADC_DAT                    ;名刺(会社)のデータ検出設定
    CALL  !STA_SEA                ;先頭ページの検出
    if_bit(ADEPDF)                ;データあり
      if([HL]==B) (A)             ;検索データ一致
        MEMBANK=M_BANK (A)       ;メモリ・アドレスを登録
        MEMPAG=M (A)
        MODE2=#2                  ;データ修正モードに設定
        SET1  MD_CG3F
        SET1  MD_CG3F
      else
        MODE1=#0                  ;名刺(会社)モード選択に設定
        SET1  MD_CG1F
      endif
    else
      MODE1=#0                    ;名刺(会社)モード選択に設定
      SET1  MD_CG1F
    endif
  endif
endif
ends
ENDIF
RET

;*****
; 名刺(会社)名前選択モード
;*****
AC_CTI:
AC_STI:
  if_bit(MD_CG2F)
    CLR1  MD_CG2F
    CLR1  CR_DSPF
    CALL  !NAM_DSP
    FOR_BNK=#0
    FOR_ADR=#ADRSTA
    REV_BNK=#0
    REV_ADR=#ADRSTA
    MODE3=#0
    SET1  MD_CG3F
  endif
  IF_BIT(MD_CG3F)
    CLR1  MD_CG3F
    SWITCH(MODE3)                 ;検索方向の指定
  CASE 0:
    FOR(j=#0;j<#7;j++)           ;1ページ表示数のカウント
      H=FOR_ADR (A)
      L=#00H
      M_BANK=FOR_BNK (A)         ;現在のメモリ・アドレスを設定
    
```

```

CALL  IBANKSET
B=#ADC_DAT           ;名刺(会社)データ参照
C=#16                ;名刺(会社)会社名の行設定
CALL  ISEA_FOR
if_bit(ADSEAF)
  D=H (A)             ;転送元アドレスの設定
  E=#16
  PUSH DE             ;アドレス過渡
  switch(j)           ;転送先アドレスの設定
  case 0:              ;2行目に設定
    CALL  INEX_ADR
    CALL  INAM_DSP
    HL=#LCDDS+30
    break
  case 1:              ;3行目に設定
    HL=#LCDDS+60
    break
  case 2:              ;4行目に設定
    HL=#LCDDS+90
    break
  case 3:              ;5行目に設定
    HL=#LCDDS+120
    break
  case 4:              ;6行目に設定
    HL=#LCDDS+150
    break
  case 5:              ;7行目に設定
    HL=#LCDDS+180
    break
  case 6:              ;8行目に設定
    HL=#LCDDS+210
  ends
  POP  DE
  CALL  IDTILSET
  H=D (A)
  CALL  IADR_INC
  FOR_ADR=H (A)
  FOR_BNK=M_BANK (A)
elseif_bit(ADENDF)   ;検索データなし
  break
endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)   ;1ページ表示数のカウント
  H=REV_ADR (A)
  L=#00H
  M_BANK=REV_BNK (A) ;現在のメモリ・アドレスを設定
  CALL  IBANKSET
  B=#ADC_DAT         ;名刺(会社)データ参照
  C=#16              ;名刺(会社)会社名の行設定
  CALL  ISEA_REV
  if_bit(ADSEAF)
    D=H (A)           ;転送元アドレスの設定
    E=#16
    PUSH DE           ;アドレス過渡
    switch(j)         ;転送先アドレスの設定
    case 0:           ;8行目に設定
      CALL  IBEF_ADR
      CALL  INAM_DSP
      HL=#LCDDS+210
      break
    case 1:           ;7行目に設定
      HL=#LCDDS+180
      break
    case 2:           ;6行目に設定
      HL=#LCDDS+150
      break
    case 3:           ;5行目に設定
      HL=#LCDDS+120
      break
    case 4:           ;4行目に設定
      HL=#LCDDS+90
      break
    case 5:           ;3行目に設定

```

```

                HL=#LCDDS+60
                break
            case 6:                                ; 2行目に設定
                HL=#LCDDS+30
            ends
            POP    DE
            CALL  !DT1LSET
            H=D (A)
            CALL  !ADR_DEC
            REV_ADR=H (A)
            REV_BNK=M_BANK (A)
        elseif_bit(ADENDF)                        ; 検索データなし
            break
        endif
    NEXT
ENDS
INV_DS=#2
SETI  IV_CHGF
ENDIF

IF_BIT(KY_SIGF)                                ; キー入力あり
CLR1  KY_SIGF
switch(KEY_CD)
case CUR_UP:                                    ; 反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SETI  IV_CHGF                            ; 反転表示行変更状態に設定
    endif
    break
case CUR_DN:                                    ; 反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)
        case 2:                                  ; 転送先アドレスの設定
            HL=#LCDDS+60
            break
        case 3:                                  ; 4行目に設定
            HL=#LCDDS+90
            break
        case 4:                                  ; 5行目に設定
            HL=#LCDDS+120
            break
        case 5:                                  ; 6行目に設定
            HL=#LCDDS+150
            break
        case 6:                                  ; 7行目に設定
            HL=#LCDDS+180
            break
        case 7:                                  ; 8行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!=#SPC) (A)                        ; 次の行に名前あり
            INV_DS++
            SETI  IV_CHGF                            ; 反転表示行変更状態に設定
        endif
    endif
    break
case CUR_RI:
    MODE3=#1
    SETI  MD_CG3F
    break
case CUR_LE:
    MODE3=#0
    SETI  MD_CG3F
    break
case CLEAR:
    MODE2=#0
    SETI  MD_CG1F
    break
case ENTER:
    CLR1  ADSEAF
    M_BANK=#0                                    ; 外部メモリを開始位置に設定
    CALL  !BANKSET
    H=#ADRSTA
    L=#0
    B=#ADC_DAT                                    ; 比較データを名刺(会社)データに設定

```

```

C=#16 ;1ライン比較の開始アドレスを設定(HL+C)
switch(INV_DS) ;検索対象行アドレスの設定
case 2: ;2行目に設定
    DE=#LCDDS+30
    break
case 3: ;3行目に設定
    DE=#LCDDS+60
    break
case 4: ;4行目に設定
    DE=#LCDDS+90
    break
case 5: ;5行目に設定
    DE=#LCDDS+120
    break
case 6: ;6行目に設定
    DE=#LCDDS+150
    break
case 7: ;7行目に設定
    DE=#LCDDS+180
    break
case 8: ;8行目に設定
    DE=#LCDDS+210
ends

if([DE]!="#SPC) (A)
    repeat
        CALL ISAVADR ;メモリ・アドレスを設定
        if_bit(CY)
            CALL IADR_INC
        endif
        if([HL]==#VAC_DAT) (A) ;検索アドレスのデータは空きデータ
            break ;登録データなし→検索終了
        else
            MODE2=#2 ;登録データあり→データ表示
            SETI MD_CG3F
        endif
        until_bit(ICMPERF)

        MEMBNK=M_BANK (A)
        MENPAG=N (A)
    endif
    break
case UP:
    if(SEACHR<#'Z') ;検索文字'A'-'Y'
        SETI MD_CG2F
        SEACHR++ ;次の文字に変更
    endif
    break
case DOWN:
    if(SEACHR>#'A') ;検索文字'B'-'Z'
        SETI MD_CG2F
        SEACHR-- ;前の文字に変更
    endif
ends
ENDIF
RET

;*****
; 名刺(会社)名前修正モード
;*****
AC_CCOR:
    if_bit(MD_CG3F) ;修正データの表更
        CLRI MD_CG3F
        M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
        CALL IBANKSET
        D=MENPAG (A)
        E=#16
        HL=#LCDDS ;LCD表示アドレスの設定
        CALL IDTIPSET ;表示データの転送
        CUR_ADR=#30 ;カーソル・アドレスを1文字目に設定
        SETI CR_DSPF
        INV_DS=#0
        SETI IV_CHGF
        SETI LCDCRGF
    endif

```

```

IF_BIT(KY_SIGF)
CLR1 KY_SIGF
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
  switch(KEY_CD)
  case DELETE:
    B=CUR_ADR (A) ;現在のカーソル・アドレスを設定
    X=CUR_ADR (A) ;桁数の設定
    A=#0
    C=#30 ;1行の文字数
    AX/=C
    X=C (A) ;桁数
    C=CUR_ADR (A) ;データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(X<#29) (A)
      [HL+B]=[HL+C] (A) ;データ転送
      B++
      C++
      X++
    endwhile
    [HL+B]=#SPC (A) ;行最終アドレスにスペース・データを設定
    break
  case CLEAR: ;前モードに設定
    MODE2=#0
    SET1 MD_CG1F
    break
  case ENTER: ;修正データの登録
    M_BANK=MEMBANK (A)
    CALL !BANKSET
    H=MEMPAG (A)
    L=#16
    DE=#LCDDS
    CALL !DT1PSET
    MODE1=#0
    SET1 MD_CG1F
    break
  case S_DEL: ;1ライン・データ削除
    A=CUR_ADR ;カーソル・アドレスの取り込み
    X=#0H
    A<->X
    C=#30 ;1行文字数
    AX/=C ;行数算出(X)
    A=#30
    AX*=X ;行先頭アドレスの設定
    AX+=#LCDDS
    HL=AX ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL !DT1LSET
    CUR_ADR=X (A) ;カーソル・アドレスを先頭に設定
    break
  case CUR_RI:
    if(CUR_ADR<#239) ;カーソル・アドレス1行目最終行以内
      CUR_ADR++
    endif
    break
  case CUR_LE:
    if(CUR_ADR>#30) ;カーソル・アドレス2行目先頭アドレス以上
      CUR_ADR--
    endif
    break
  case CUR_UP:
    if(CUR_ADR>=#60) ;カーソル・アドレスを1行上(+30)に設定
      CUR_ADR-=#30
    endif
    break
  case CUR_DN:
    if(CUR_ADR<#210) ;カーソル・アドレスを1行下(-30)に設定
      CUR_ADR+=#30
    endif
    break
  case UP:
    M_BANK=MEMBANK (A) ;メモリ・アドレスの設定
    CALL !BANKSET
    H=MEMPAG (A) ;表示データのページ設定
    L=#0

```

```

if(H==#ADREND) (A)           ;バンク最終アドレス
if(M_BANK<#3)                ;バンク0-バンク2
M_BANK++                     ;バンク更新
CALL !BANKSET
H=#ADRSTA
if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
SETI MD_CG3F
MEMPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H++                           ;次データ参照
if([HL]==#ADC_DAT) (A)
SETI MD_CG3F ;アドレスの更新
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
break
case DOWN:
M_BANK=MEMBNK (A)           ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG (A)                ;表示データのページ設定
L=#0
if(H==#ADRSTA) (A)         ;バンク先頭アドレス
if(M_BANK>#0)              ;バンク1-バンク3
M_BANK--                   ;バンク更新
CALL !BANKSET
H=#ADREND
if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
SETI MD_CG3F
MEMPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H--                           ;次データ参照
if([HL]==#ADC_DAT) (A)
SETI MD_CG3F ;アドレスの更新
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
ends
ELSE
B=CUR_ADR (A)               ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)          ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#239)          ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
ENDIF
RET

;*****
; 名刺(会社)データ検索モード
;*****
AC_SEA:
if_bit(MD_CG1F)
CLR! MD_CG1F
INV_DS=#0 ;反転表示行削除
SETI IV_CHGF
HL=#LCDDS ;メニュー表示データの設定
DE=#NA_SEA
CALL !DTILSET ;LCD1行目のデータを設定

DE=#CLR1LD
CALL !DTILSET ;LCD2行目のデータを設定

DE=#ATOM_D
CALL !DTILSET ;LCD3行目のデータを設定

DE=#CLR1LD

```

```

CALL !DTILSET ;LCD 4行目のデータを設定

DE=#NTOZ_D
CALL !DTILSET ;LCD 5行目のデータを設定

DE=#CLRILD
CALL !DTILSET ;LCD 6行目のデータを設定

DE=#NEXT_P
CALL !DTILSET ;LCD 7行目のデータを設定

DE=#CLRILD
CALL !DTILSET ;LCD 8行目のデータを設定

MODE2=#0
MODE3=#0
MODE4=#0

SETI CR_DSPF ;カーソル表示の設定
CUR_ADR=#62 ;カーソル・アドレスを'A'の位置に設定
SETI LCDCHGF

endif
switch(MODE2)
case 0:
CALL !AC_SSEL ;先頭文字選択モード
break
case 1:
CALL !AC_STI ;名前選択モード
break
case 2:
CALL !AC_SSEA ;データ表示モード
break
case 3:
CALL !AC_OFV ;会社員一覧表示
ends
RET

;*****
; 名刺(会社)名前検索モード
;*****
AC_SSEA:
if_bit(COMM_F) ;通信状態
CALL !S_CONN ;通信処理
if_bit(!COMM_F) ;通信終了
SETI MD_CG3F ;通信前の状態に設定
endif
else
if_bit(MD_CG3F) ;修正データの変更
CLR1 MD_CG3F
M_BANK=MEMBNK(A) ;メモリ・アドレスの設定
CALL !BANKSET
D=MEMPAG(A)
E=#16
HL=#LCDDS ;LCD表示アドレスの設定
CALL !DTIPSET ;表示データの転送
CLR1 CR_DSPF ;カーソル削除
INV_DS=#0 ;反転表示行削除
SETI !V_CHGF
SETI LCDCHGF
endif
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD)
case CLEAR: ;前モードに設定
SETI MD_CG1F
break
case UP:
M_BANK=MEMBNK(A) ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG(A) ;表示データのページ設定
L=#0
if(H=#ADREND)(A) ;バンク最終アドレス
if(M_BANK<#3) ;バンク0-バンク2
M_BANK++ ;バンク更新
CALL !BANKSET

```

```

H=#ADRSTA
if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
    SET1 MD_CG3F
    MEMPAG=H (A) ;アドレスの更新
    MEMBNK=M_BANK (A)
endif
endif
else
H++ ;次データ参照
if([HL]==#ADC_DAT) (A)
    SET1 MD_CG3F ;アドレスの更新
    MEMPAG=H (A)
    MEMBNK=M_BANK (A)
endif
endif
break
case DOWN:
M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
CALL !BANKSET
H=MEMPAG (A) ;表示データのページ設定
L=#0
if(H==#ADRSTA) (A) ;バンク先頭アドレス
    if(M_BANK>#0) ;バンク0-バンク2
        M_BANK-- ;バンク更新
        CALL !BANKSET
    H=#ADREND
    if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
        SET1 MD_CG3F
        MEMPAG=H (A) ;アドレスの更新
        MEMBNK=M_BANK (A)
    endif
endif
else
H-- ;次データ参照
if([HL]==#ADC_DAT) (A)
    SET1 MD_CG3F ;アドレスの更新
    MEMPAG=H (A)
    MEMBNK=M_BANK (A)
endif
endif
break
case COMM:
TRN_ADR=MEMPAG (A) ;データ送信アドレスの設定
TRN_BNK=MEMBNK (A)
COM_VOL=#1 ;データ送信数iバイトの設定
SET1 COMM_F
SET1 TR_CG1F
break
case ENTER:
NODE2=#3 ;社員一覧表示モードに設定
NODE3=#0
SET1 MD_CG3F
BACPAG=MEMPAG (A) ;メモリ・アドレスのバックアップ
BACBNK=MEMBNK (A)
FOR_BNK=#0
REV_BNK=#0
FOR_ADR=#ADRSTA
REV_ADR=#ADRSTA
CALL !OFV_DSP
ends
ENDIF
endif
RET

;*****
; 名刺(会社)社員一覧表示モード
;*****
AC_OFW:
IF_BIT(MD_CG3F)
CLR1 MD_CG3F
SWITCH(NODE3) ;検索方向の指定
CASE 0:
FOR(j=#0;j<#7;j++) ;1ページ表示数のカウント
H=FOR_ADR (A)
L=#00H

```



```

M_BANK=FOR_BNK (A)           ;現在のメモリ・アドレスを設定
CALL !BANKSET
DE=#LCDDS                    ;会社名アドレス設定
B=#ADM_DAT                   ;名刺(個人)データ参照
C=#16+150                    ;名刺(個人)会社名の行設定
CALL !SEA_FIL
if_bit(ADSEAF)
  D=H (A)                    ;転送元アドレスの設定
  E=#16
  PUSH DE
  switch(j)                  ;転送先アドレスの設定
  case 0:                    ;2行目に設定
    CALL !NEX_ADR
    CALL !OFV_DSP
    HL=#LCDDS+30
    break
  case 1:                    ;3行目に設定
    HL=#LCDDS+60
    break
  case 2:                    ;4行目に設定
    HL=#LCDDS+90
    break
  case 3:                    ;5行目に設定
    HL=#LCDDS+120
    break
  case 4:                    ;6行目に設定
    HL=#LCDDS+150
    break
  case 5:                    ;7行目に設定
    HL=#LCDDS+180
    break
  case 6:                    ;8行目に設定
    HL=#LCDDS+210
  ends
  POP DE
  CALL !DT1LSET
  H=D (A)
  CALL !ADR_INC
  FOR_ADR=H (A)
  FOR_BNK=M_BANK (A)
elseif_bit(ADENDF)          ;検索データなし
  break
endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)         ;1ページ表示数のカウント
  H=REV_ADR (A)
  L=#00H
  M_BANK=REV_BNK (A)       ;現在のメモリ・アドレスを設定
  CALL !BANKSET
  DE=#LCDDS                ;会社名アドレス設定
  B=#ADM_DAT               ;名刺(個人)データ参照
  C=#16+150                ;名刺(個人)会社名の行設定
  CALL !SEA_RIL
  if_bit(ADSEAF)
    D=H (A)                ;転送元アドレスの設定
    E=#16
    PUSH DE
    switch(j)              ;転送先アドレスの設定
    case 0:                ;8行目に設定
      CALL !BBF_ADR
      CALL !OFV_DSP
      HL=#LCDDS+210
      break
    case 1:                ;7行目に設定
      HL=#LCDDS+180
      break
    case 2:                ;6行目に設定
      HL=#LCDDS+150
      break
    case 3:                ;5行目に設定
      HL=#LCDDS+120
      break
    case 4:                ;4行目に設定

```

```

                HL=#LCDDS+90
                break
            case 5:                                ;3行目に設定
                HL=#LCDDS+60
                break
            case 6:                                ;2行目に設定
                HL=#LCDDS+30
            ends
            POP    DE
            CALL  !DTILSET
            H=D (A)
            CALL  !ADR_DEC
            REV_ADR=H (A)
            REV_BNK=M_BANK (A)
        elseif_bit(ADENDF)                        ;検索データなし
            break
        endif
    NEXT
ENDS
ENDIF

if_bit(KY_SIGF)
CLR1  KY_SIGF
switch(KEY_CD)
case CLEAR:                                     ;前モードに設定
    MODE2=#2
    SET1  MD_CG3F
    MEMPAG=BACPAG (A)
    MEMBNK=BACBNK (A)
    break
case CUR_RI:                                    ;会社員一覧を次ページに設定
    SET1  MD_CG3F
    MODE3=#0
    break
case CUR_LE:                                    ;会社員一覧を前ページに設定
    SET1  MD_CG3F
    MODE3=#1
ends
endif
RET

;*****
;  名刺(会社)先頭文字選択モード
;*****
AC_DEL:
    if_bit(MD_CG1F)
        CLR1  MD_CG1F
        INV_DS=#0                                ;反転表示行削除
        SET1  IV_CHGF
        HL=#LCDDS                                ;メニュー表示データの設定
        DE=#NA_SEA
        CALL  !DTILSET                            ;LCD1行目のデータを設定

        DE=#CLR1LD
        CALL  !DTILSET                            ;LCD2行目のデータを設定

        DE=#ATOM_D
        CALL  !DTILSET                            ;LCD3行目のデータを設定

        DE=#CLR1LD
        CALL  !DTILSET                            ;LCD4行目のデータを設定

        DE=#NTOZ_D
        CALL  !DTILSET                            ;LCD5行目のデータを設定

        DE=#CLR1LD
        CALL  !DTILSET                            ;LCD6行目のデータを設定

        DE=#NEXT_P
        CALL  !DTILSET                            ;LCD7行目のデータを設定

        DE=#CLR1LD
        CALL  !DTILSET                            ;LCD8行目のデータを設定

    MODE2=#0

```

```

MODE3=#0
MODE4=#0

SET1  CR_DSPF          ;カーソル表示の設定
CUR_ADR=#62          ;カーソル・アドレスを'A'の位置に設定
SET1  LCDCHGF

endif
switch(MODE2)
case 0:
  CALL  IAC_DSEL        ;先頭文字選択モード
  break
case 1:
  CALL  IAC_DTI         ;名前選択モード
  break
case 2:
  CALL  IAC_DDEL        ;データ表示モード
ends
RET

;*****
;  名刺(会社)名前選択モード(削除)
;*****
AC_DTI:
  if_bit(MD_CG2F)
  CLR1  MD_CG2F
  SET1  MD_CG3F
  CLR1  CR_DSPF
  CALL  !NAM_DSP
  FOR_BNK=#0
  FOR_ADR=#ADRSTA
  REV_BNK=#0
  REV_ADR=#ADRSTA
  MODE3=#0
  SET1  MD_CG3F
endif
IF_BIT(MD_CG3F)
CLR1MD_CG3F
SWITCH(MODE3)          ;検索方向の指定
CASE 0:
  FOR(j=#0;j<#7;j++)   ;1ページ表示数のカウント
  H=FOR_ADR(A)
  L=#00H
  M_BANK=FOR_BNK(A)    ;現在のメモリ・アドレスを設定
  CALL  !BANKSET
  B=#ADC_DAT
  C=#16
  CALL  !SEA_FOR
  if_bit(ADSEAF)
  D=H(A)                ;転送元アドレスの設定
  E=#16
  PUSH  DE              ;アドレス退避
  switch(j)             ;転送先アドレスの設定
  case 0:               ;2行目に設定
    CALL  !NEX_ADR
    CALL  !NAM_DSP
    HL=#LCDDS+30
    break
  case 1:               ;3行目に設定
    HL=#LCDDS+60
    break
  case 2:               ;4行目に設定
    HL=#LCDDS+90
    break
  case 3:               ;5行目に設定
    HL=#LCDDS+120
    break
  case 4:               ;6行目に設定
    HL=#LCDDS+150
    break
  case 5:               ;7行目に設定
    HL=#LCDDS+180
    break
  case 6:               ;8行目に設定
    HL=#LCDDS+210
  ends

```

```

        POP     DE
        CALL   !DT1LSET
        H=D (A)
        CALL   !ADR_INC
        FOR_ADR=H (A)
        FOR_BNK=M_BANK (A)
        elseif_bit(ADENDF)           ;検索データなし
        break
    endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)                 ;1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)              ;現在のメモリ・アドレスを設定
    CALL   !BANKSET
    B=#ADC_DAT
    C=#16
    CALL   !SEA_REV
    if_bit(ADSEAF)
        D=H (A)                     ;転送元アドレスの設定
        E=#16
        PUSH  DE                     ;アドレス過渡
        switch(j)                    ;転送先アドレスの設定
        case 0:                       ;8 行目に設定
            CALL   !BEP_ADR
            CALL   !NAM_DSP
            HL=#LCDDS+210
            break
        case 1:                       ;7 行目に設定
            HL=#LCDDS+180
            break
        case 2:                       ;6 行目に設定
            HL=#LCDDS+150
            break
        case 3:                       ;5 行目に設定
            HL=#LCDDS+120
            break
        case 4:                       ;4 行目に設定
            HL=#LCDDS+90
            break
        case 5:                       ;3 行目に設定
            HL=#LCDDS+60
            break
        case 6:                       ;2 行目に設定
            HL=#LCDDS+30
        ends
        POP     DE
        CALL   !DT1LSET
        H=D (A)
        CALL   !ADR_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
        elseif_bit(ADENDF)           ;検索データなし
        break
    endif
NEXT
ENDS
INV_DS=#2
SETI   IV_CHGF
ENDIF

IF_BIT(KY_SIGF)                     ;キー入力あり
CLR!   KY_SIGF
switch(KEY_CD)
case CUR_UP:                         ;反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SETI   IV_CHGF              ;反転表示行変更状態に設定
    endif
    break
case CUR_DN:                         ;反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)              ;転送先アドレスの設定

```

```

case 2:                                ; 3行目に設定
    HL=#LCDDS+60
    break
case 3:                                ; 4行目に設定
    HL=#LCDDS+90
    break
case 4:                                ; 5行目に設定
    HL=#LCDDS+120
    break
case 5:                                ; 6行目に設定
    HL=#LCDDS+150
    break
case 6:                                ; 7行目に設定
    HL=#LCDDS+180
    break
case 7:                                ; 8行目に設定
    HL=#LCDDS+210
ends
if([HL]!="#SPC) (A)                    ; 次の行に名前あり
    INV_DS++
    SET1  IV_CHGF                      ; 反転表示行変更状態に設定
endif
endif
break
case CUR_RI:
    MODE3=#1
    SET1  MD_CG3F
    break
case CUR_LE:
    MODE3=#0
    SET1  MD_CG3F
    break
case CLEAR:
    MODE2=#0
    SET1  MD_CG1F
    break
case DELETE:
    BACPAG=MEMPAG (A)                  ; メモリ・アドレスのバックアップ
    BACBNK=MEMBNK (A)
    B=#ADC_DAT                          ; 名刺(会社)データの検索
    C=#16                                ; 比較アドレスの設定
    switch(INV_DS)
case 2:
    DE=#LCDDS+30
    break
case 3:                                ; 3行目に設定
    DE=#LCDDS+60
    break
case 4:                                ; 4行目に設定
    DE=#LCDDS+90
    break
case 5:                                ; 5行目に設定
    DE=#LCDDS+120
    break
case 6:                                ; 6行目に設定
    DE=#LCDDS+150
    break
case 7:                                ; 7行目に設定
    DE=#LCDDS+180
    break
case 8:                                ; 8行目に設定
    DE=#LCDDS+210
ends
if([DE]!="#SPC) (A)                    ; 検索開始アドレスの設定
    H=#ADRSTA
    L=#0
    M_BANK=#0
    CALL  !BANKSET
    CALL  !SAVADR                      ; 名刺(会社)データの検索
    CALL  !DT_DEL
    ADC_VOL--                          ; データ数1減算
endif
FOR_ADR=#ADRSTA
FOR_BNK=#0
SET1  MD_CG2F

```

```

MODE3=#0
SET1 MD_CG3F
break
case ENTER:
CLR1 ADSEAF
M_BANK=#0 ;外部メモリを開始位置に設定
CALL !BANKSET
H=#ADRSTA
L=#0
B=#ADC_DAT ;比較データを名刺(会社)データに設定
C=#16 ;1ライン比較の開始アドレスを設定(HL+C)
switch(INV_DS) ;検索対象行アドレスの設定
case 2: ;2行目に設定
DE=#LCDDS+30
break
case 3: ;3行目に設定
DE=#LCDDS+60
break
case 4: ;4行目に設定
DE=#LCDDS+90
break
case 5: ;5行目に設定
DE=#LCDDS+120
break
case 6: ;6行目に設定
DE=#LCDDS+150
break
case 7: ;7行目に設定
DE=#LCDDS+180
break
case 8: ;8行目に設定
DE=#LCDDS+210
ends

if([DE]!="#SPC) (A)
repeat
CALL !SAVADR ;メモリ・アドレスを設定
if_bit(CY)
CALL !ADR_INC
endif
if([HL]==#VAC_DAT) (A) ;検索アドレスのデータは空きデータ
break ;登録データなし→検索終了
else ;登録データあり→データ表示
MODE2=#2
SET1 MD_CG3F
endif
until_bit(ICMPERF)

MEMBNK=M_BANK (A)
MEMPAG=H (A)
endif
break
case UP:
if(SEACHR<#'Z') ;検索文字'A'-'Y'
SET1 MD_CG2F
SEACHR++ ;次の文字に変更
endif
break
case DOWN:
if(SEACHR>#'A') ;検索文字'B'-'Z'
SET1 MD_CG2F
SEACHR-- ;前の文字に変更
endif
ends
ENDIF
RET

;*****
; 名刺(会社)名前検索モード
;*****
AC_DDEL:
if_bit(MD_CG3F) ;修正データの変更
CLR1 MD_CG3F
M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
CALL !BANKSET

```

```

D=MEMPAG (A)
E=#16
HL=#LCDDS ;LCD表示アドレスの設定
CALL !DTIPSET ;表示データの転送
CLR! CR_DSPF ;カーソル表示なし
INV_DS=#0 ;反転表示削除
SET! IV_CHGF
SET! LCDCHGF
endif
IF_BIT(KY_SIGF)
CLR! KY_SIGF
switch(KEY_CD)
case CLEAR: ;前モードに設定
    NODE2=#0
    SET! MD_CG1F
    break
case UP:
    M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
    CALL !BANKSET
    H=MEMPAG (A) ;表示データのページ設定
    L=#0
    if(H==#ADREND) (A) ;バンク最終アドレス
        if(M_BANK<#3) ;バンク0-バンク2
            M_BANK++ ;バンク更新
            CALL !BANKSET
            H=#ADRSTA
            if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
                SET! MD_CG3F
                MEMPAG=H (A) ;アドレスの更新
                MEMBNK=M_BANK (A)
            endif
        endif
    else
        H++ ;次データ参照
        if([HL]==#ADC_DAT) (A)
            SET! MD_CG3F ;アドレスの更新
            MEMPAG=H (A)
            MEMBNK=M_BANK (A)
        endif
    endif
    break
case DOWN:
    M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
    CALL !BANKSET
    H=MEMPAG (A) ;表示データのページ設定
    L=#0
    if(H==#ADRSTA) (A) ;バンク先頭アドレス
        if(M_BANK>#0) ;バンク0-バンク2
            M_BANK-- ;バンク更新
            CALL !BANKSET
            H=#ADREND
            if([HL]==#ADC_DAT) (A) ;次データ=名刺(会社)データ
                SET! MD_CG3F
                MEMPAG=H (A) ;アドレスの更新
                MEMBNK=M_BANK (A)
            endif
        endif
    else
        H-- ;次データ参照
        if([HL]==#ADC_DAT) (A)
            SET! MD_CG3F ;アドレスの更新
            MEMPAG=H (A)
            MEMBNK=M_BANK (A)
        endif
    endif
    break
case DELETE:
    H=MEMPAG (A)
    L=#00H
    M_BANK=MEMBNK (A)
    CALL !BANKSET
    CALL !DT_DEL
    ADC_VOL--
    NODE2=#0
    SET! MD_CG1F

```

```
ends
ENDIF
RET

;*****
; 名刺（会社）名前表示設定
;*****
NAM_DSP:
CALL !LCD_DSC
HL=#LCDDS
DE=#NAM_DAT ;1行目の表示データを設定
CALL !DT1LSET
!LCDDS+11=SEACHR (A) ;検索文字の表示
SETI LCDCHGF
RET

;*****
; 名刺（会社）会社員表示設定
;*****
OFW_DSP:
CALL !LCD_DSC ;表示クリア
HL=#LCDDS ;会社名を1行目に設定
D=BACPAG (A)
E=#16
M_BANK=BACBNK (A)
CALL !BANKSET
CALL !DT1LSET
SETI LCDCHGF
RET
END
```


4.6 カレンダー・モード

カレンダー・モードでは、モード選択や、カレンダー表示（1901-2099年）、スケジュール・データの登録、修正、削除、休日の設定を行っています。

4.6.1 登録データのフォーマット

カレンダー・モードでは、スケジュール・データと、指定休日データの2種類の登録データがあります。各登録フォーマットを図4-11、図4-12に示します。

図4-11 スケジュール・データの登録フォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																													
xx00H	04	年月日			開始終了時間			注	00	00	00	00	00	00	00	00																													
xx10H	データの種類																																												
xx20H																データ検索用パラメータ																													
xx30H																															LCD表示データ240バイト (30文字×8行)														
xx40H																																													
xx50H																																													
xx60H																																													
xx70H																																													
xx80H																																													
xx90H																																													
xxA0H																																													
xxB0H																																													
xxC0H																																													
xxD0H																																													
xxE0H																																													
xxF0H																																													

注 アラーム・オン：00H
アラーム・オフ：FFH

図4 - 12 指定休日の登録フォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xx00H	05	年月日	00	00	00	00	00	00	00	00	00	00	00	00	00	00
xx10H	05	年月日	00	00	00	00	00	00	00	00	00	00	00	00	00	00
xx20H	05	年月日	00	00	00	00	00	00	00	00	00	00	00	00	00	00
⋮																
xxF0H	05	年月日	00	00	00	00	00	00	00	00	00	00	00	00	00	00

4.6.2 モード選択条件

カレンダー・モードの遷移状態の条件を表4 - 7に示します。

表4 - 7 カレンダー・モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード		
3	0	-	-	-	カレンダー・モード選択		
	1	1	0	-	現時刻のカレンダー表示モード		
			1	0	-	スケジュール・モード選択	
				1	0	-	開始時間入力モード
					1	-	終了時間入力モード
					2	-	アラーム設定モード
					3	-	用件入力モード
				4	-	メモ入力モード	
				2	0	-	開始時間修正モード
					1	-	終了時間修正モード
			2		-	アラーム修正モード	
			3		-	用件修正モード	
			2	0	0	-	カレンダー表示年データの設定モード
					1	-	カレンダー表示月データの設定モード
			3	3	1	-	指定月のカレンダー表示モード
	0	-			-	休日設定のモード選択	
		-			-	週休の設定モード	
		0			-	指定休日の年入力モード	
	2	1			-	指定休日の月入力モード	
		2			-	指定休日の日入力モード	

備考 - : don't care

4.6.3 カレンダー表示処理

カレンダー表示では、次の内容を表示しています。

カレンダーの月表示
 スケジュールの日付指定表示
 スケジュール・マーク表示
 週休表示
 指定休日表示

(1) カレンダーの月表示

カレンダーの月表示では、1901年1月1日(火)の曜日データ(2:火曜日)を基準とし、表示する月の1日の曜日を算出しています。算出方法は、次のとおりです。まずはじめに年数による曜日の移動数と、平年/うるう年の月による曜日の移動数から表示する月の1日の曜日を算出しています。そのあと、表示月の日数を決定しています。

年数による曜日の移動数

平年のときは、曜日が1年で1日移動します。ただし、4年に1回のうるう年のときは、さらに1日移動します。したがって、次のような計算方法により、年数による曜日の移動数を求めることができます。

$$(\text{指定年} - 1901) + (\text{指定年} - 1901) / 4$$

月による曜日の移動数

指定した月の1日の曜日が、その年の1月1日の曜日に対してどれだけ移動しているかを、データ・テーブル(CALE_T)に格納しておきます。したがって、月による曜日の移動数は、平年、うるう年の判断と月のデータにより、データを参照しています。データ構成としては、上位4ビットをうるう年における移動数に、下位4ビットを平年における移動数に設定しています。

<月による曜日移動数>

CALE_T:

DB	00H	; 1月
DB	33H	; 2月
DB	43H	; 3月
.	.	.
.	.	.
.	.	.
DB	65H	; 12月

指定月の1日の曜日

または の結果に1901年1月1日の曜日データ(2:火曜日)を加算し、1週間(7)で割った余りが、指定月の1日の曜日になります。

例 1992年3月

$$(1992 - 1901) + (1992 - 1901) / 4 = 113$$

$$\text{うるう年3月の移動数} = 4$$

$$\begin{aligned} \text{指定月の1日の曜日} &= (113 + 4 + 2) \div 7 \\ &= 17 \cdots 0 \end{aligned}$$

0:日曜日

カレンダー表示データの設定

カレンダーの月表示は、空白部分を含めたデータのテーブルを用意しています。したがって、 の結果より、テーブルの先頭アドレスを決定します。

<カレンダー表示テーブル>

	土	金	木	水	火	月	曜日による 開始アドレス
CALDSP:							
DB	SPC,SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC						
DB 日	SPC, '1', SPC, '2', SPC, '3', SPC, '4', SPC, '5', SPC, '6'						
DB	'2', '6', '2', '7', '2', '8', '2', '9', '3', '0', '3', '1'						
DB	SPC,SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC						
DB	SPC,SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC						

カレンダー表示の最終日の設定

表示月最終日は、平年、うるう年の最終日のデータをテーブルに用意しています。カレンダー表示の場合は、 で設定したデータに対して、表示月の最終日以降を空白にする処理を行っています。

(2) スケジュールの日付指定表示

スケジュールの日付指定は、(xx)の位置になります。()を表示するアドレスは、指定日と、表示月1日の曜日データからアドレスを決定します。

(3) スケジュール・マーク表示

スケジュールが登録されている日付の左脇に‘*’マークを表示します。処理手順を次に示します。

データ登録メモリの $\times \times 00\text{H}$ のデータが 04H のアドレスを検索します。

表示月の年月データと、 で検索したデータの $\times \times 01\text{H}$, $\times \times 02\text{H}$ のデータが一致しているアドレスを検索します。

で検索したアドレスの日データ ($\times \times 03\text{H}$) を取り込み、1日の位置より表示しているアドレスを算出します。

算出したアドレスに‘*’データを書き込みます。

図4 - 13 スケジュール・データのフォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\times \times 00\text{H}$	04	年月日			開始終了時間			注	00	00	00	00	00	00	00	00

注 アラーム・オン : 00H

アラーム・オフ : FFH

(4) 週休表示

週休表示は、週休データ (WEKHOLD) により、設定しています。週休データは、1つの曜日に対して、1ビットのデータを持っています。対応するビットが0のときは平日とし、1のときは休日としています。処理手順を次に示します。

週休データをビット単位でテストします。

休日データ (ビットが1) のときは、LCDモジュールの表示エリアのアドレスをグラフィック・エリアに設定し、ドット・データ (FFH) により縦1ラインにデータを書き込みます。

図4 - 14 週休データ

	7	6	5	4	3	2	1	0
WEKHOLD	-	土	金	木	水	火	月	日

0 : 平日, 1 : 休日

(5) 指定休日表示

指定休日表示は、指定休日として登録されている部分の日付を反転表示します。処理手順を次に示します。

データ登録メモリの $\times \times 00\text{H}$ のデータが 05H のアドレスを検索します。

表示月の年月データと、 で検索したデータの $\times \times \times 1\text{H}$ 、 $\times \times \times 2\text{H}$ のデータが一致しているアドレスを検索します。

で検索したアドレスの日データ ($\times \times \times 3\text{H}$) を取り込み、1日の位置より表示しているアドレスを算出します。

LCDモジュールの表示エリアのアドレスをグラフィック・エリアに設定し、ドット・データ (FFH) により1文字分のデータを書き込みます。

4.6.4 スケジュール登録処理

スケジュール・モードは、表示月の () で示している日付が対象となり、ENTキー入力によりモード変更をしています (MODE2 = 1)。そして、REGISTERが選択されたときに登録モードとなります (MODE3 = 1)。

スケジュール登録モードでは、ENTキー入力によりMODE4の値を更新し、CLRキー入力のときは前モードに戻すためMODE3を0に設定します。登録データは、文字データと、 , , , のキー入力により、カーソルのアドレス (CUR_ADR) に文字データを書き込みます。すべてのデータ入力後 (メモ入力後のENTキー)、次の順で登録処理を行います。

メモリ最終ページの先頭アドレスの確認をします。

バンク3 (P61, P60 = 1, 1) のBF00H番地がFFH (空きエリア) 以外のときはメモリ・オーパフローにします (MODE2 = 6に設定)。

データを登録するアドレスを検索します。

検索条件: $\times \times 00\text{H}$ が 04H 以上のとき

検索したアドレスにデータを登録します。表示データ (240文字) は、 $\times \times 10\text{H}$ - $\times \times \text{FFH}$ に登録し、 $\times \times 00\text{H}$ に 04H を、 $\times \times 01\text{H}$ - $\times \times 03\text{H}$ に年月日を、 $\times \times 04\text{H}$ - $\times \times 07\text{H}$ にスケジュールの開始/終了時間のデータを、 $\times \times 08\text{H}$ にアラームのオン/オフ (00H/FFH) データを、 09H - $\times \times 0\text{FH}$ に 00H を登録します。検索アドレス以降のデータは、次のアドレス (+ 100H) にシフトします。

4.6.5 スケジュール表示，修正処理

スケジュール・モードは，表示月の（ ）で示している日付が対象となり，ENTキー入力によりモード変更をしています（MODE2=1）。そして，開始/終了時間と用件の表示された行が選択されたときに表示，修正モードとなります（MODE3=2）。選択されたデータの検索は，次の手順で行っています。

データ登録メモリの $\times \times 00\text{H}$ が 04H のデータを検索します。

で検索したアドレスの $\times \times 01\text{H}$ - $\times \times 03\text{H}$ と，対象日付のデータが一致しているアドレスを検索します。

で検索したアドレスの $\times \times 04\text{H}$ - $\times \times 07\text{H}$ と，選択された行の開始/終了時間が一致しているアドレスを検索します。

スケジュール表示，修正モードでは， ， キー入力によりMODE4の値を更新し，修正データは，文字データと ， ， ， のキー入力により，カーソルのあるアドレス（CUR_ADR）に文字データを書き込みます。

修正モードでは，ENTキー入力により修正完了とし，修正前のデータ登録メモリのアドレスに再登録します。登録後は，スケジュール・モード選択に戻します（MODE3=0）。CLRキー入力のときは，前モードに戻すためMODE3を0に設定します。

4.6.6 スケジュール削除処理

スケジュール・モードは，表示月の（ ）で示している日付が対象となり，ENTキー入力によりモード変更をしています（MODE2=1）。そして，DELETEが選択されたときにデータを削除します。データ削除を実行すると，対象となる日付のスケジュールすべてのデータを削除します。データ削除後は，現時刻のカレンダー表示モードに戻します（MODE2=0）。CLRキー入力のときは，前モードに戻すためMODE2を0に設定します。次の順でデータ削除を行っています。

データ登録メモリの $\times \times 00\text{H}$ が 04H のデータを検索します。

で検索したアドレスの $\times \times 01\text{H}$ - $\times \times 03\text{H}$ と，対象日付のデータが一致しているアドレスを検索します。

で検索したアドレスのデータを削除します。

$\times \times 00\text{H}$ が 04H の間， - の処理を繰り返します。

4.6.7 カレンダー表示月の設定処理

カレンダー表示月の設定は，年 (MODE3 = 0)，月 (MODE3 = 1) データをENTキー入力により決定します。月データ設定後は，カレンダー月表示モードにします (MODE2 = 1)。

年データ，月データは，それぞれ1バイトのデータに変換しています。

・ 1901-2099年 : 1-199(01H-C7H)

・ 1-12 : 1-12(01H-0CH)

4.6.8 指定休日の設定処理

指定休日の設定は，年(MODE3=0)，月(MODE3=1)，日(MODE3=2)データをENT，DELキー入力により決定します。

(1) 指定休日の設定

指定休日を設定するときは，データ入力後のENTキーで行います。指定休日のデータ登録の順を次に示します。

データ登録メモリの $\times \times 00H$ が $05H$ のアドレスを検索します。

現在の休日登録数 (HOL_VOL) から，登録アドレスを算出します。

図4-12のフォーマットのようにデータを登録します。

休日登録数のカウンタ (HOL_VOL) を + 1 します。

(2) 指定休日の解除

指定休日を解除するときは，データ入力後のDELキーで行います。指定休日のデータ削除の順を次に示します。

データ登録メモリの $\times \times \times 0H$ が $05H$ のアドレスを検索します。

検索したアドレスの $\times \times \times 1H$ - $\times \times \times 3H$ のデータと，指定休日のデータが一致しているアドレスを検索します。

で検索したアドレスのデータを削除します。

休日登録数のカウンタ (HOL_VOL) を - 1 します。

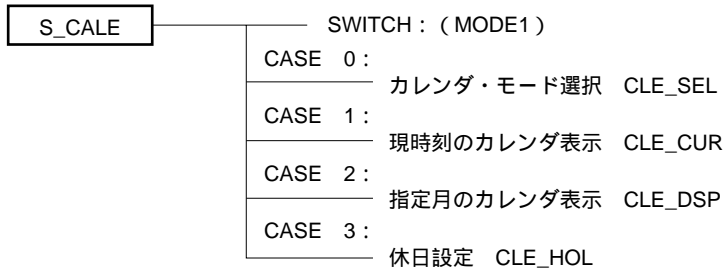
4.6.9 週休の設定処理

週休の設定処理は，モード選択されたあとに (MODE1 = 3, MODE2 = 1)，週休データ (WEKHOLD) により (図4-14参照)，各曜日のデータが1のときは 'o' を，0のときは 'x' を表示します。そして，ENTキーが入力されたときは，データを1ビットずつ取り込み週休データ (WEKHOLD) を更新します。

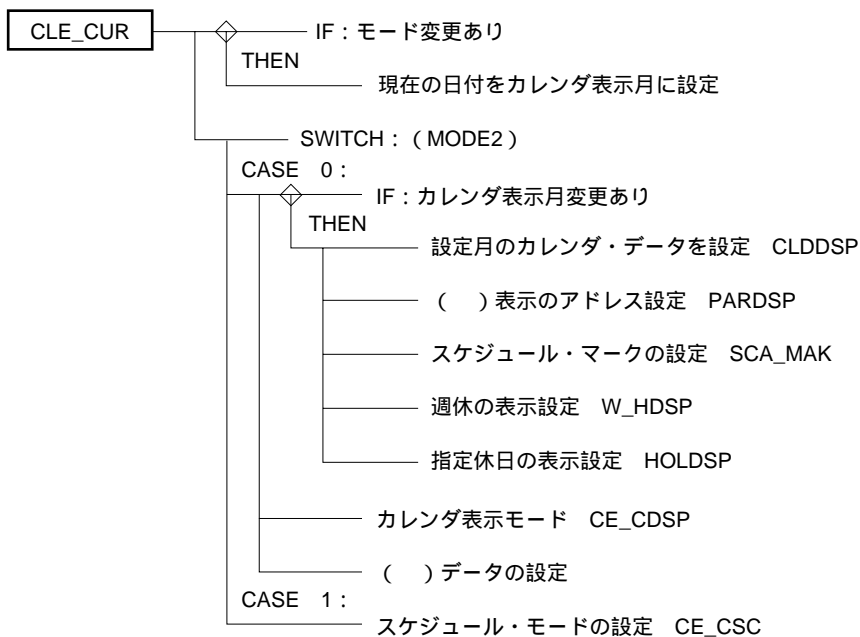
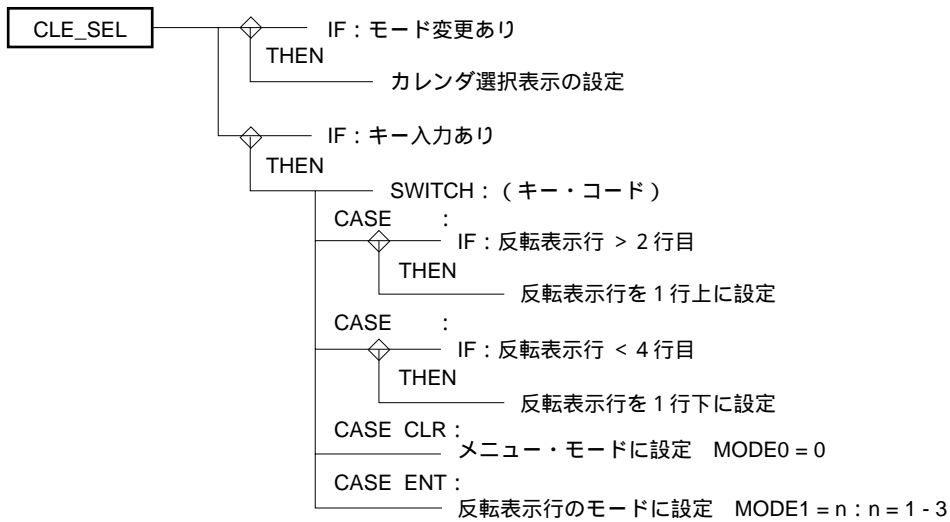
データ更新後は，カレンダー・モード選択に戻します (MODE1 = 0)。

4.6.10 SPDチャート

カレンダー・モード

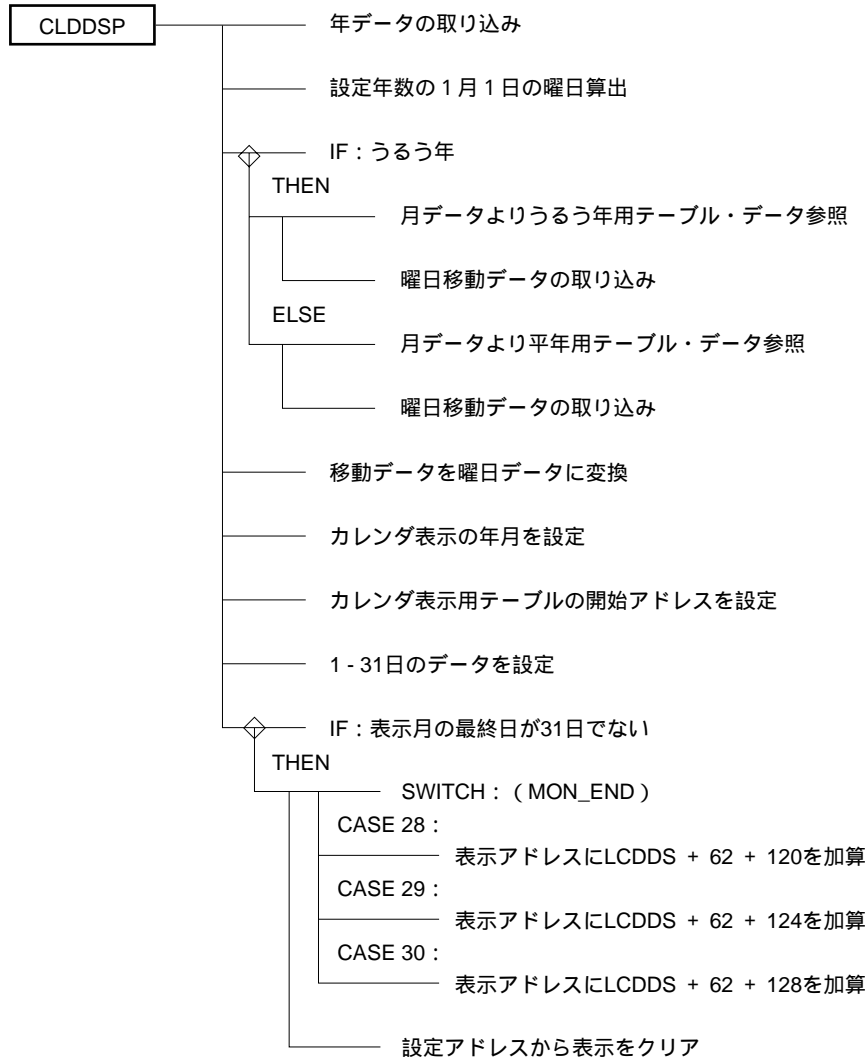


・モード選択

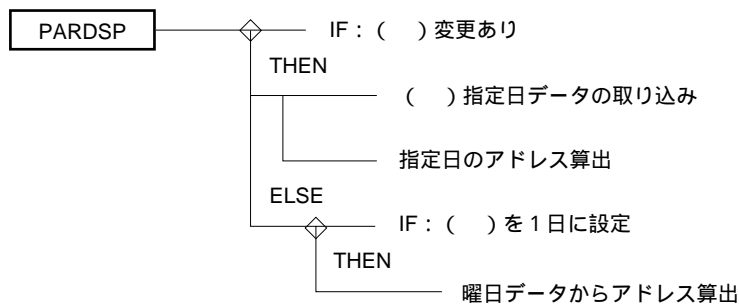


カレンダー・モード

・カレンダー表示

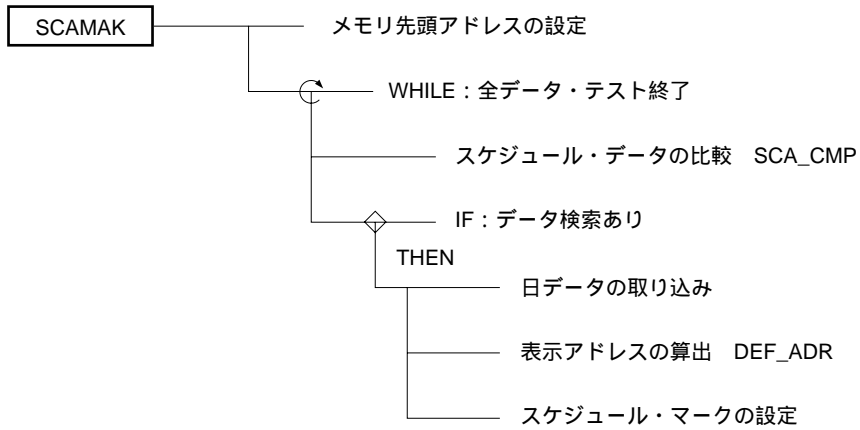


・ () アドレスの設定

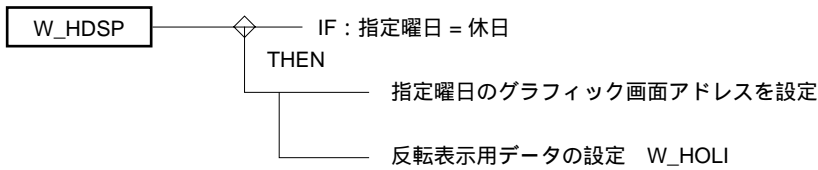


カレンダー・モード

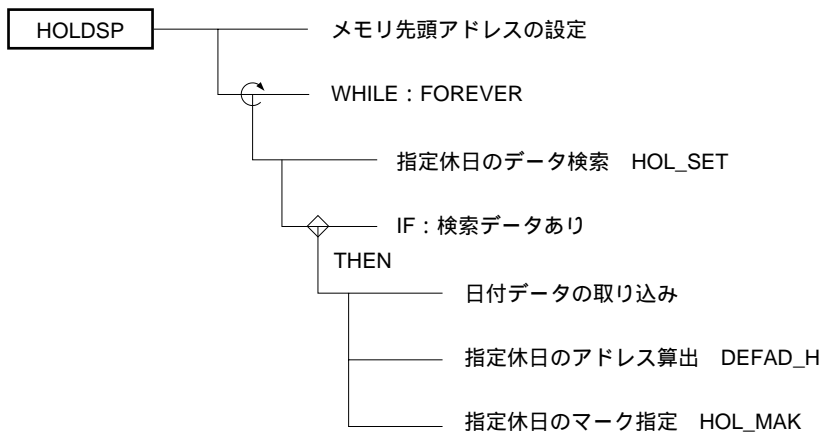
・スケジュール・マークの設定



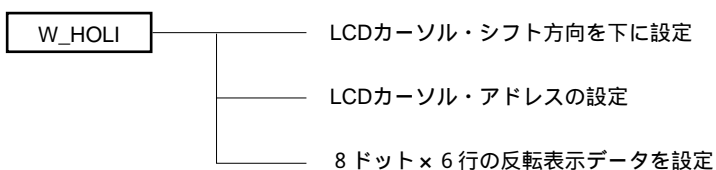
・週休表示の設定



・指定休日表示の設定

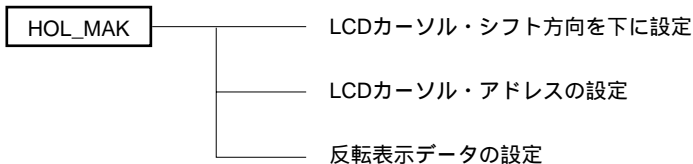


・週休表示設定

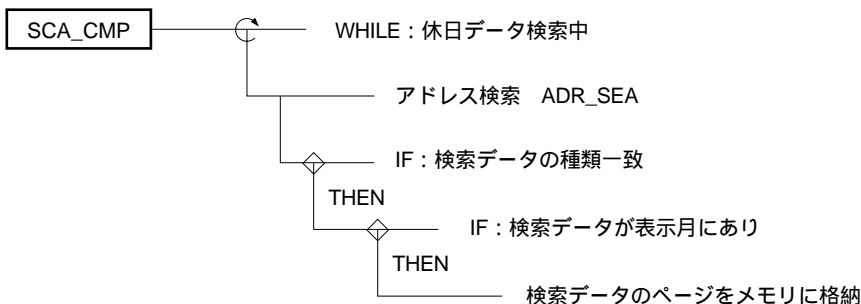


カレンダー・モード

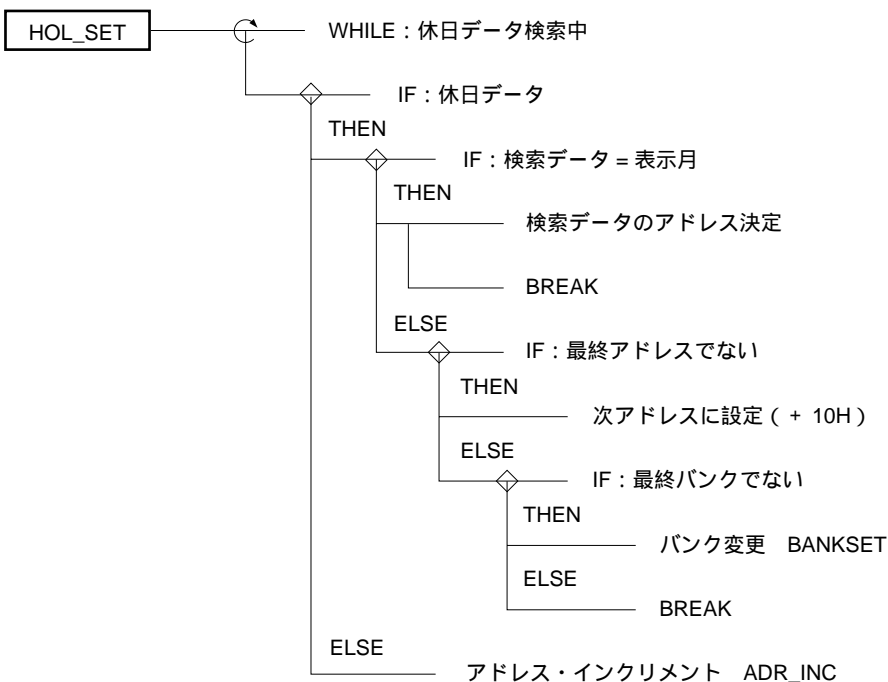
・指定休日の表示設定



・スケジュール日付比較

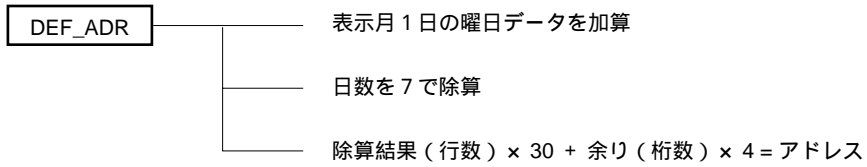


・指定休日検索

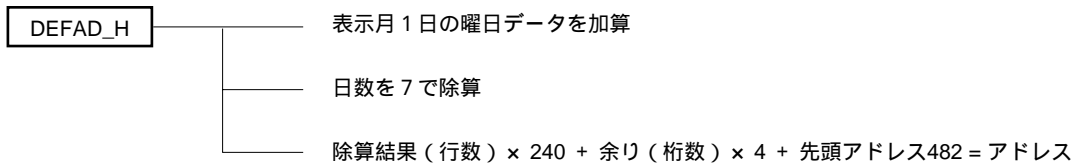


カレンダー・モード

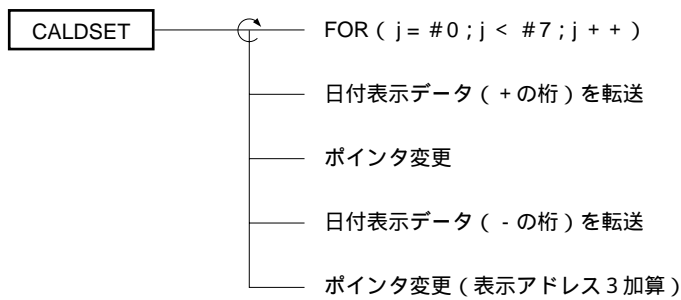
・カレンダー日付アドレス算出



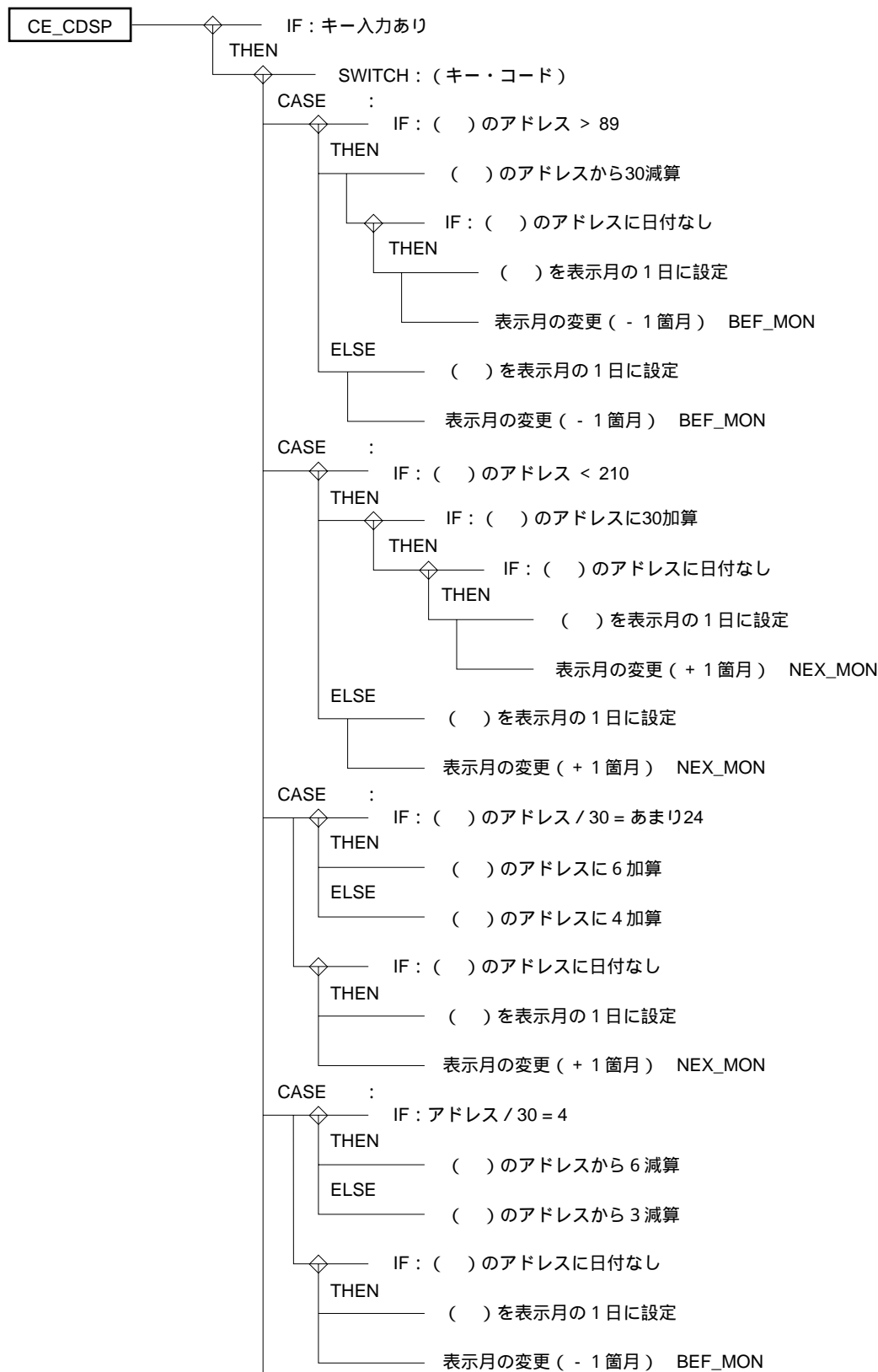
・カレンダー休日アドレス算出



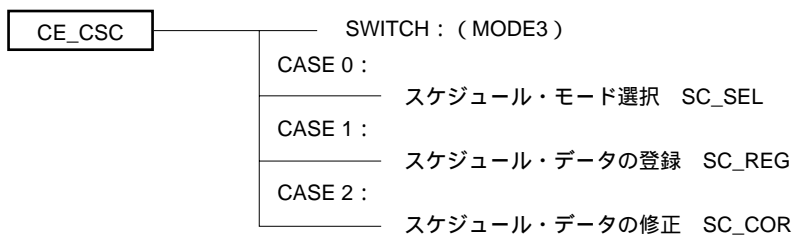
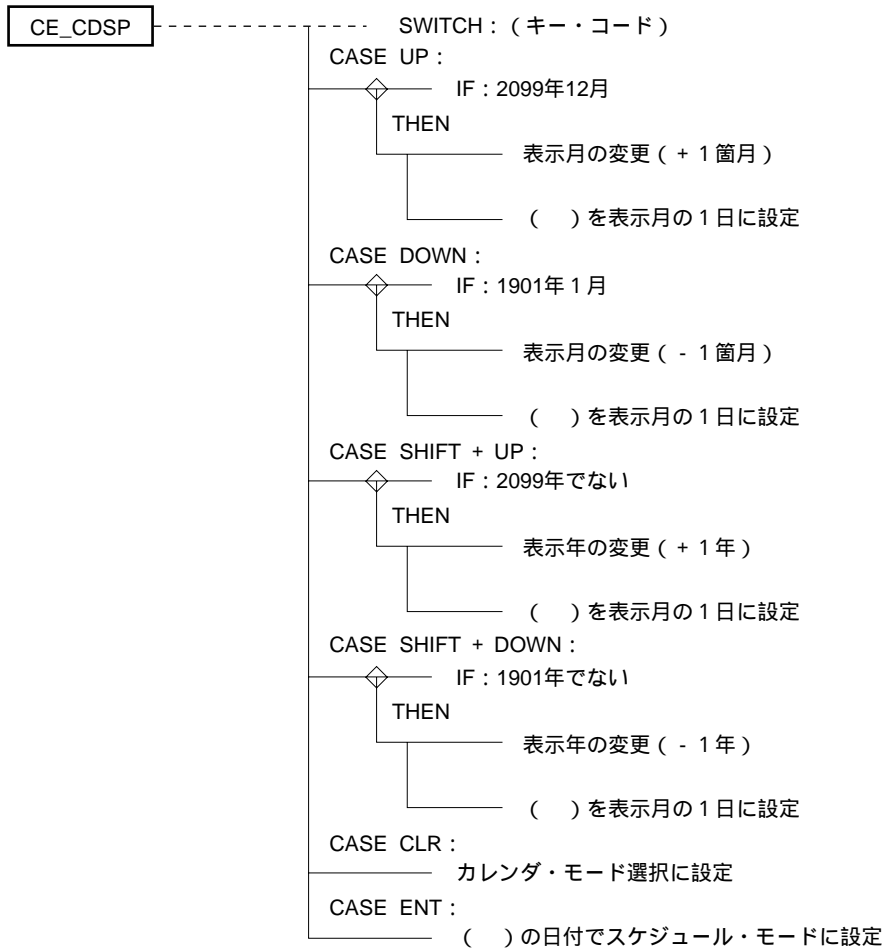
・カレンダー・データ 1 ライン設定



カレンダー・モード

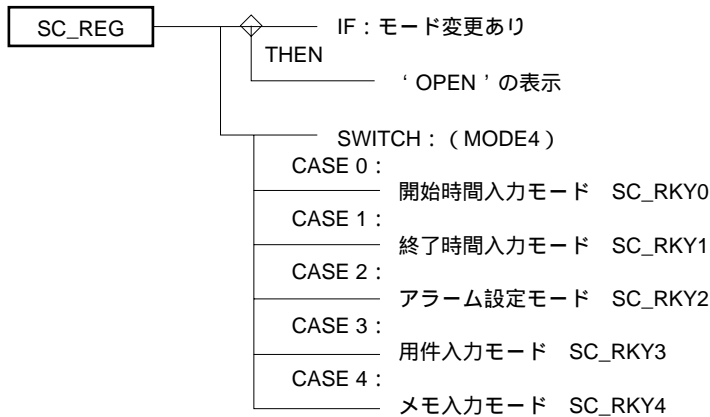
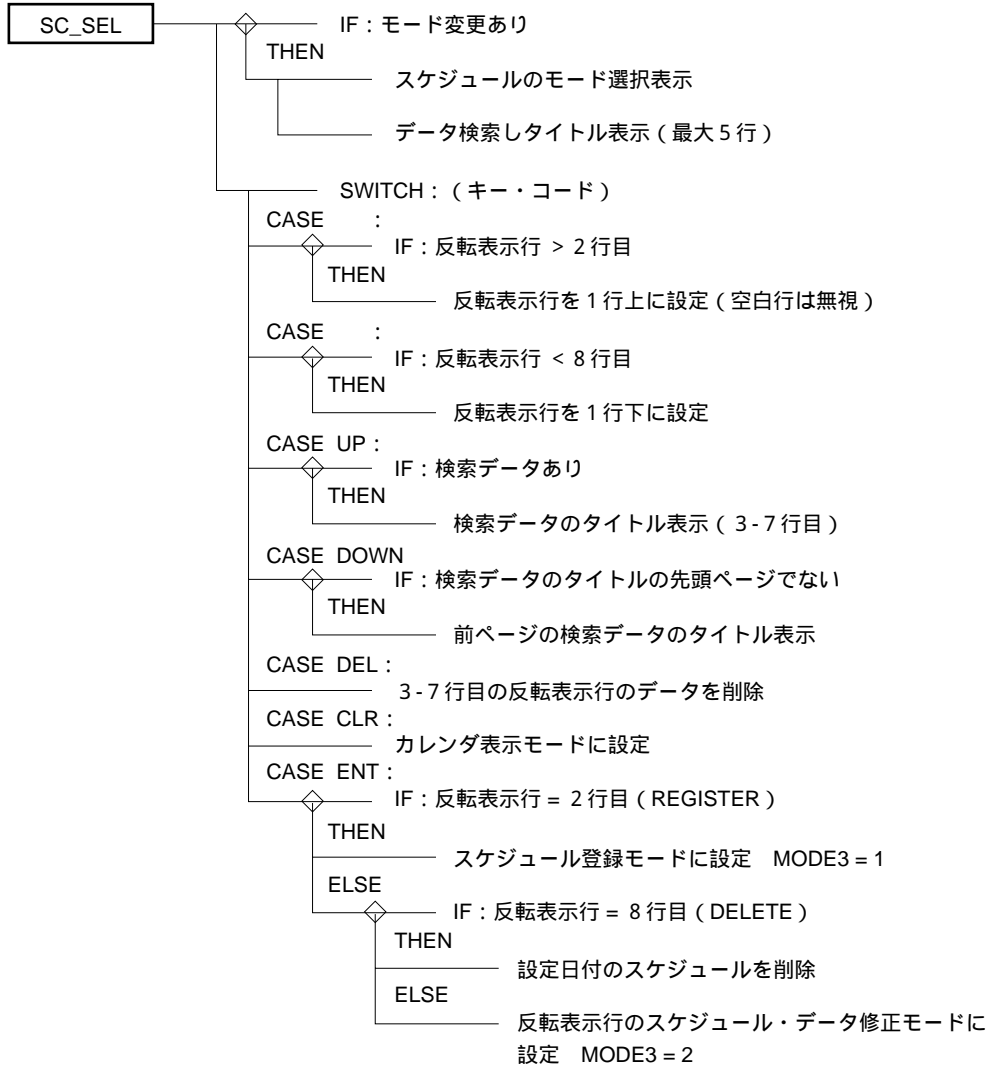


カレンダー・モード



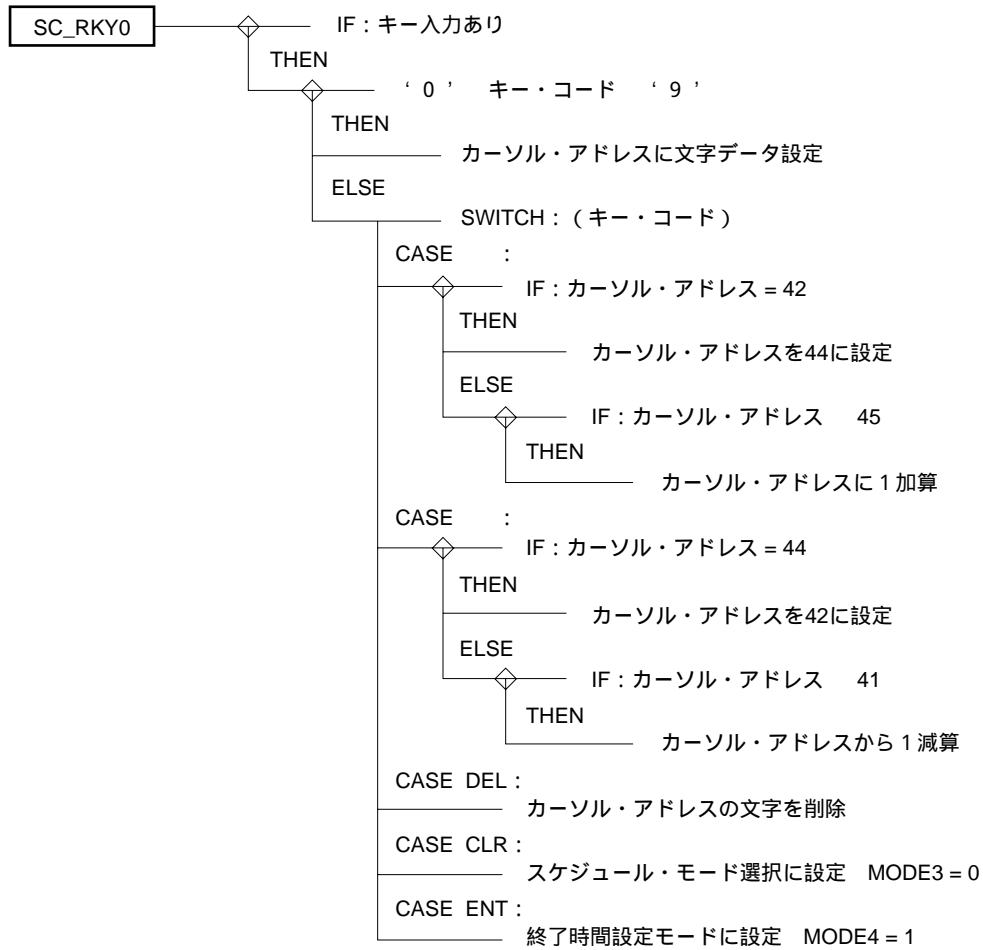
カレンダー・モード

・スケジュール登録



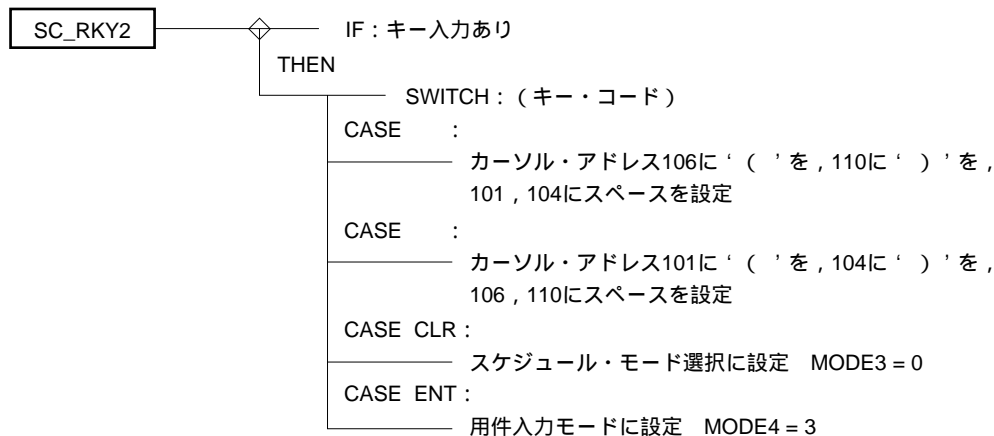
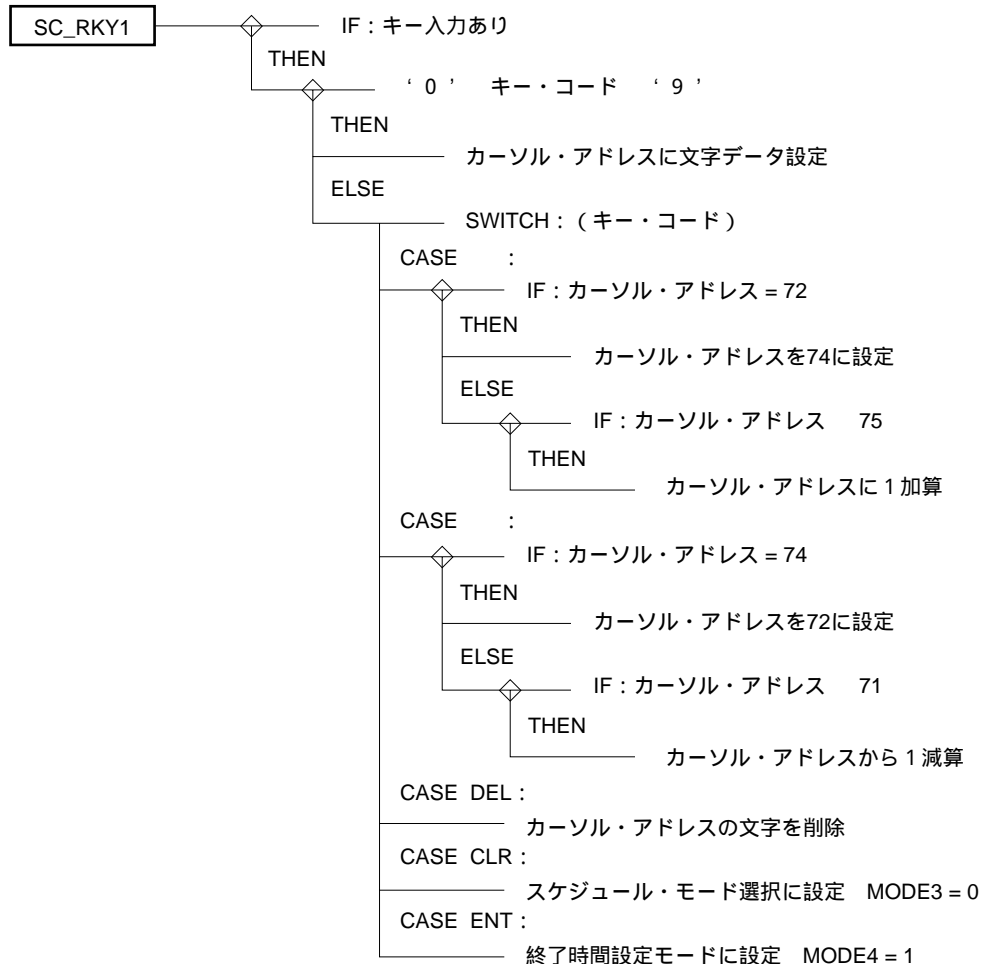
カレンダー・モード

・スケジュール登録



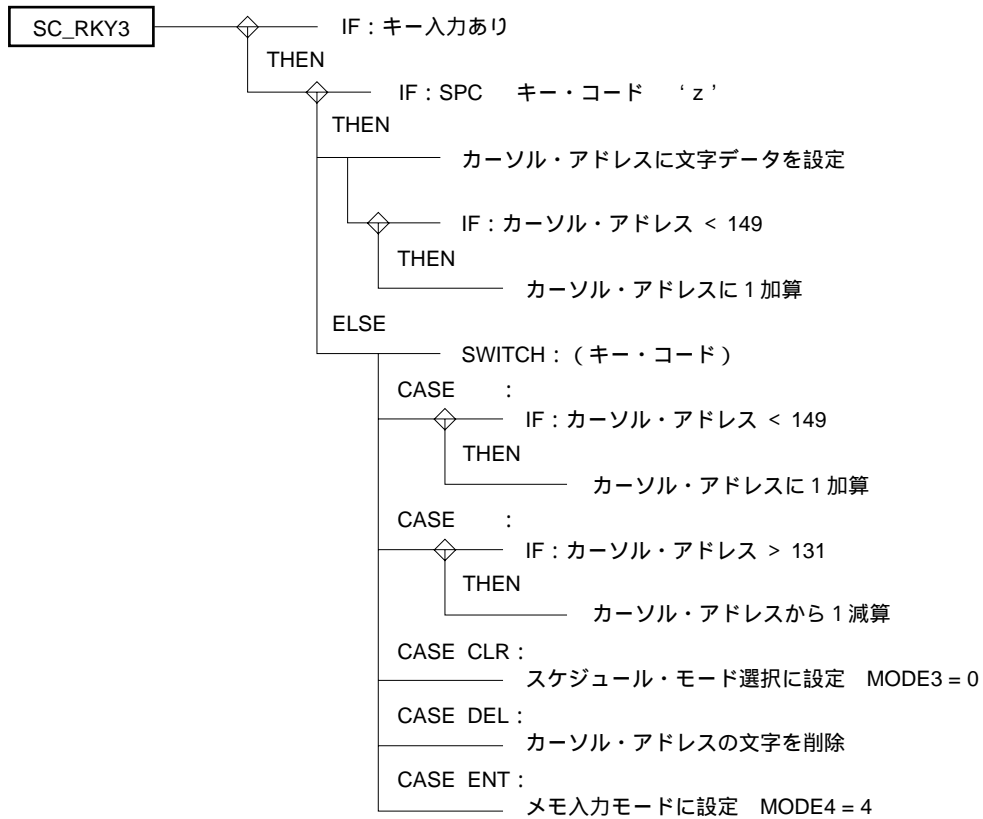
カレンダー・モード

・スケジュール登録



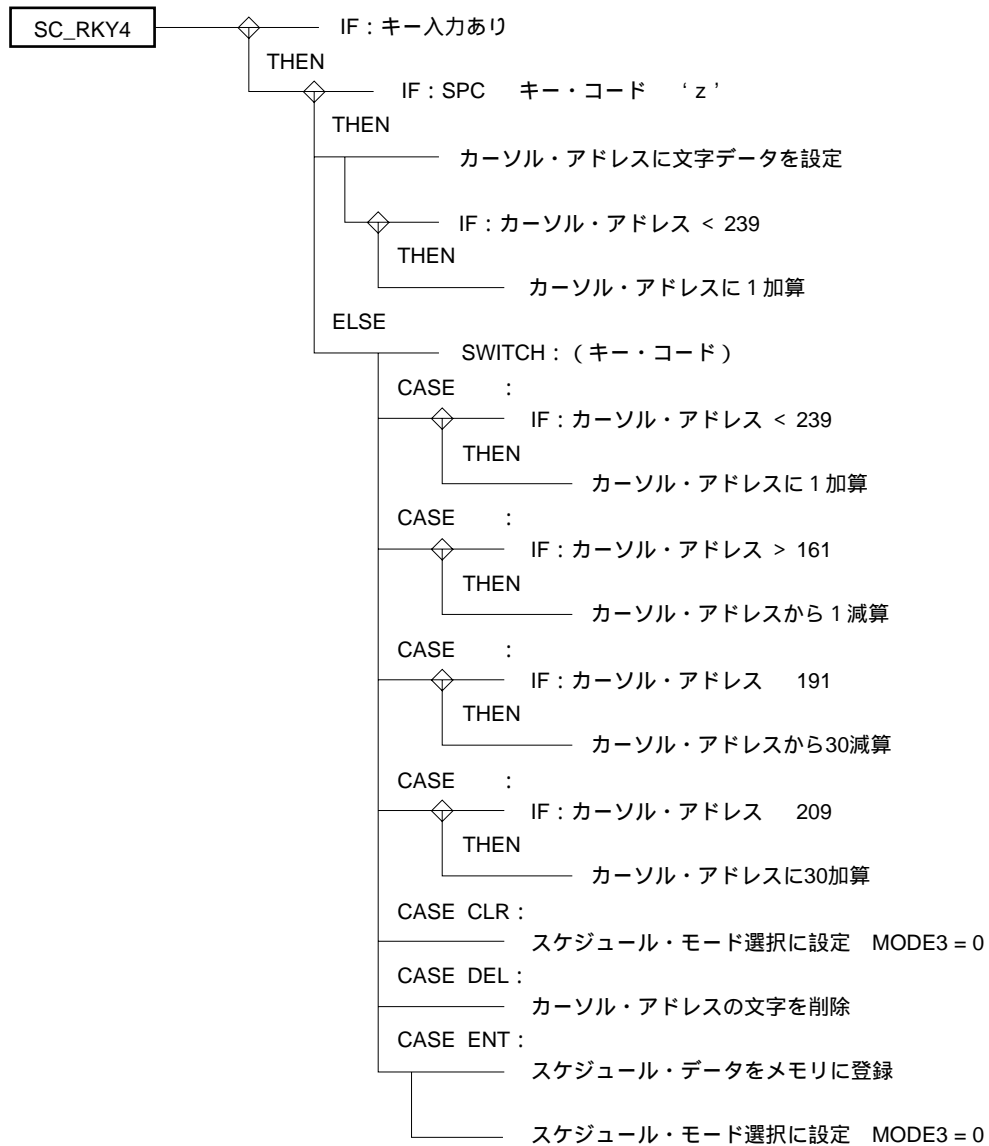
カレンダー・モード

・スケジュール登録



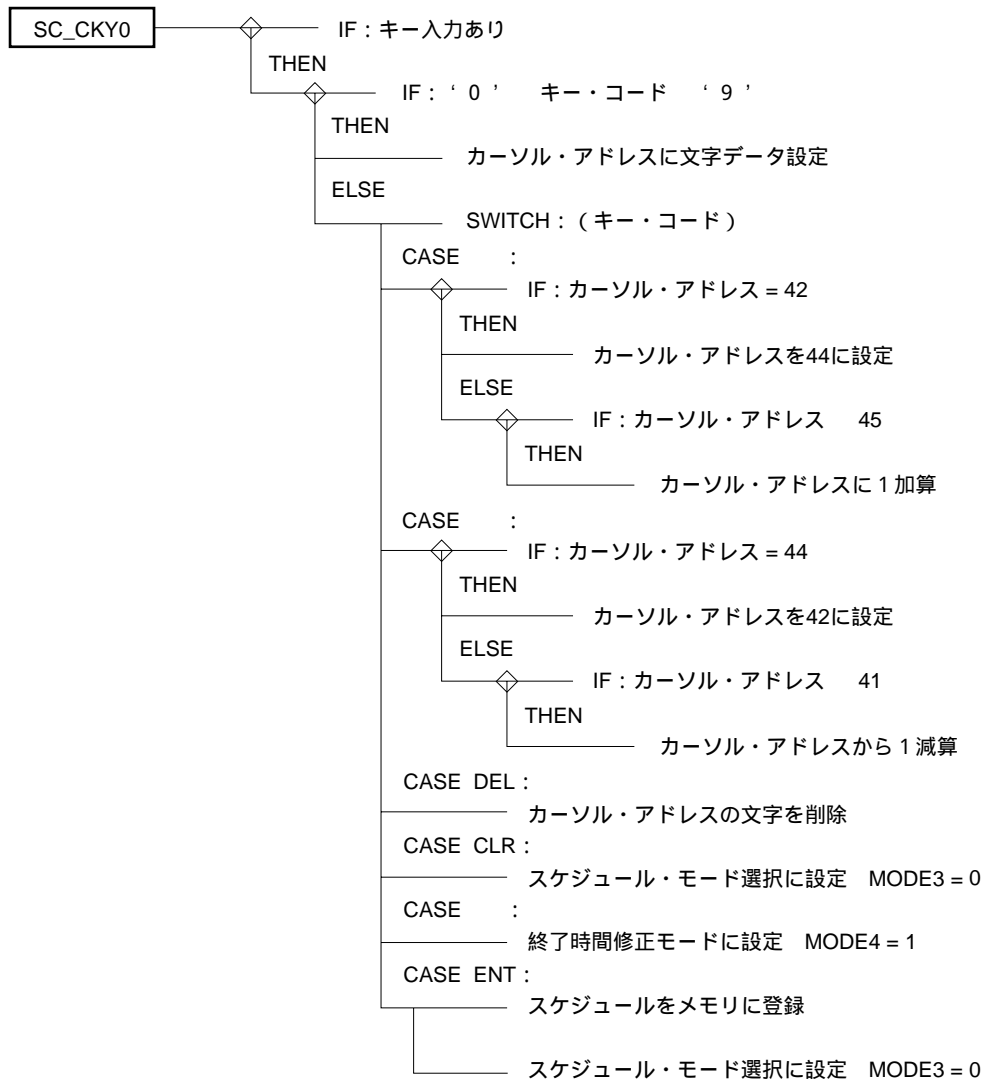
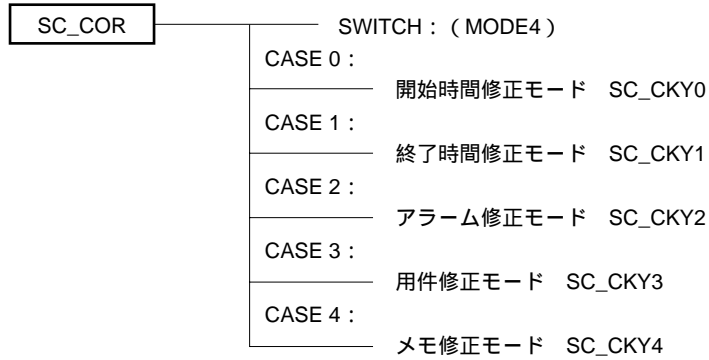
カレンダー・モード

・スケジュール登録



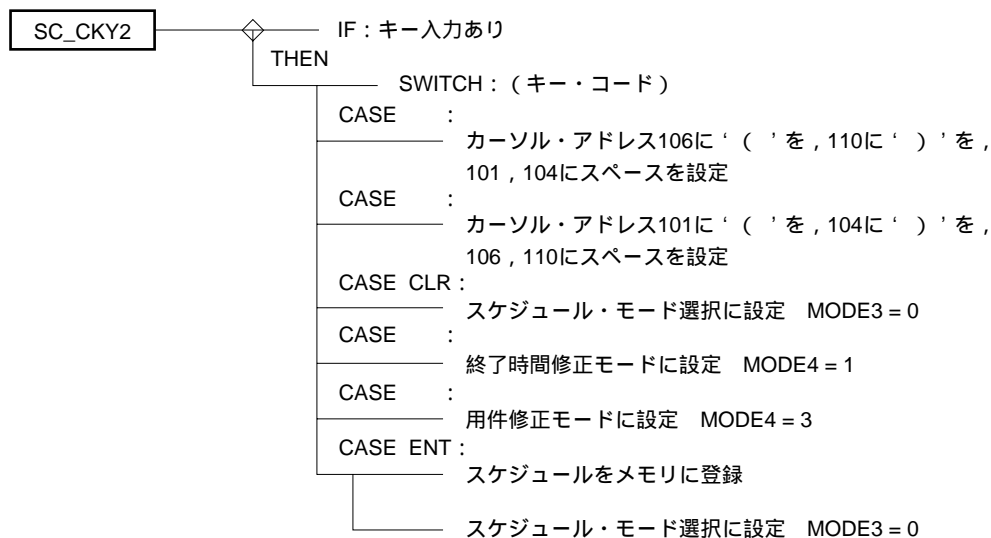
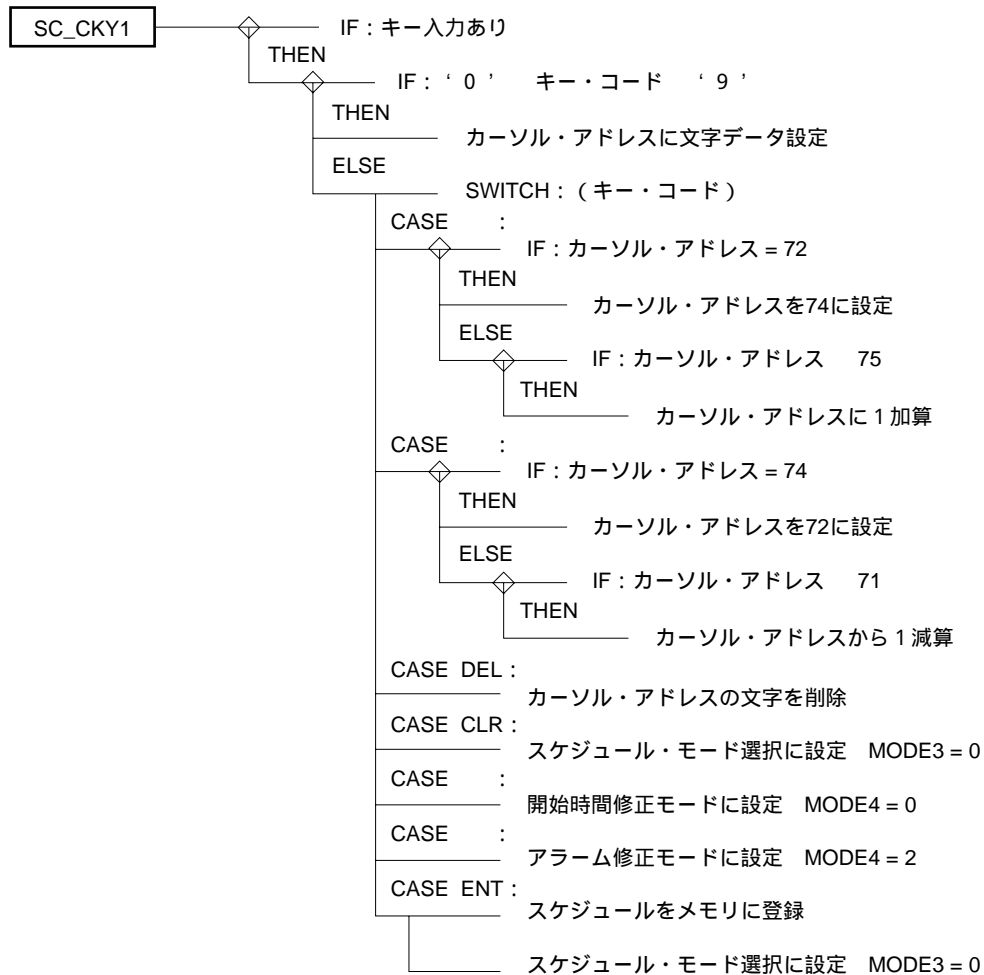
カレンダー・モード

・スケジュール修正



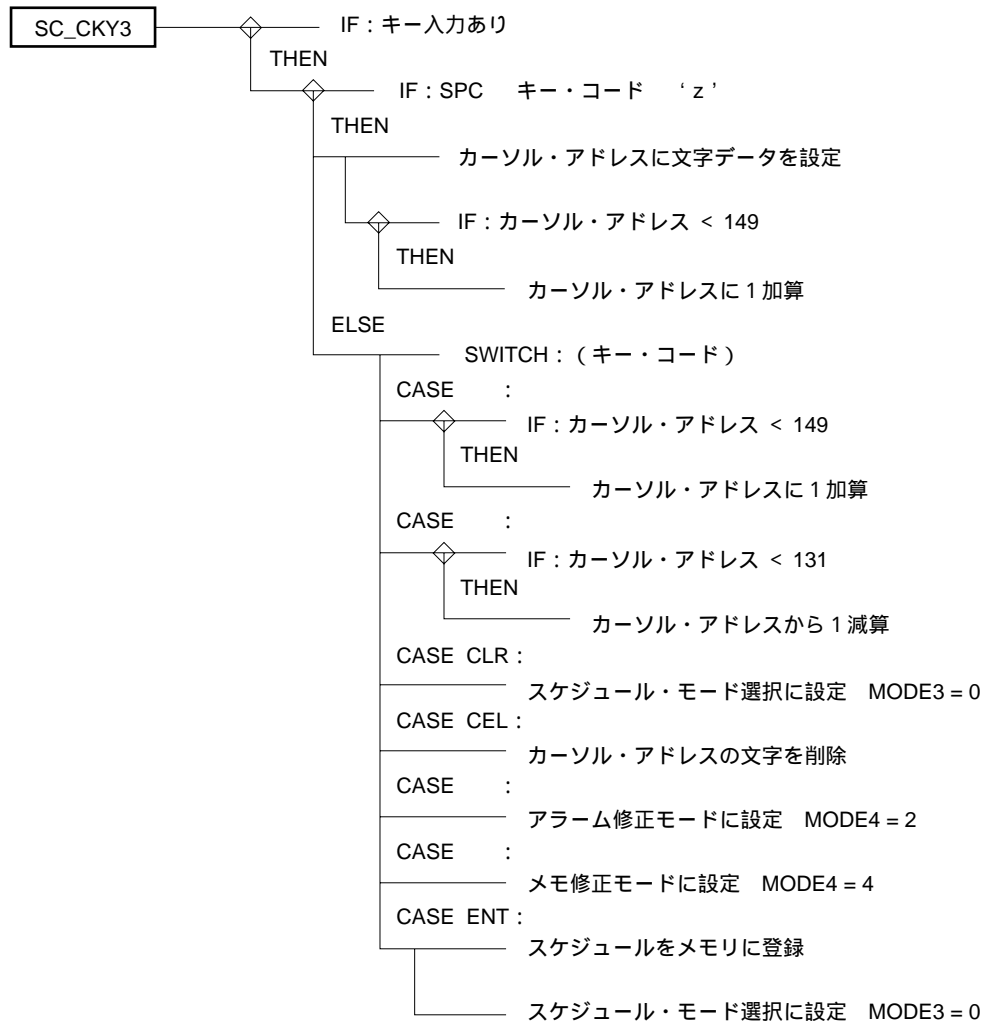
カレンダー・モード

・スケジュール修正



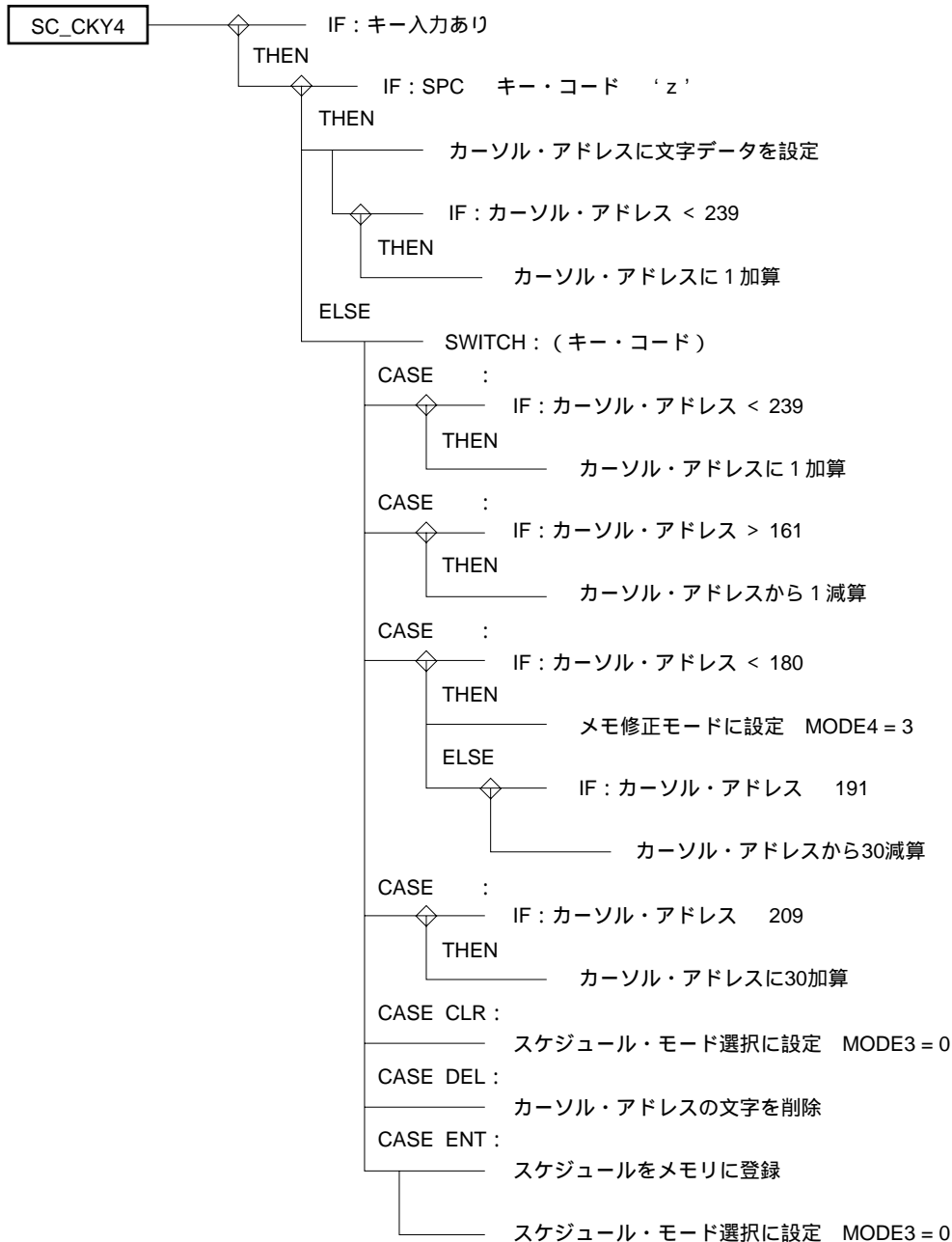
カレンダー・モード

・スケジュール修正



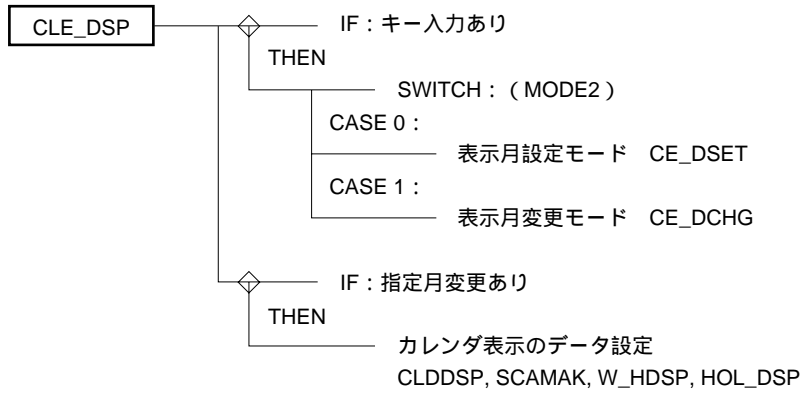
カレンダー・モード

・スケジュール登録



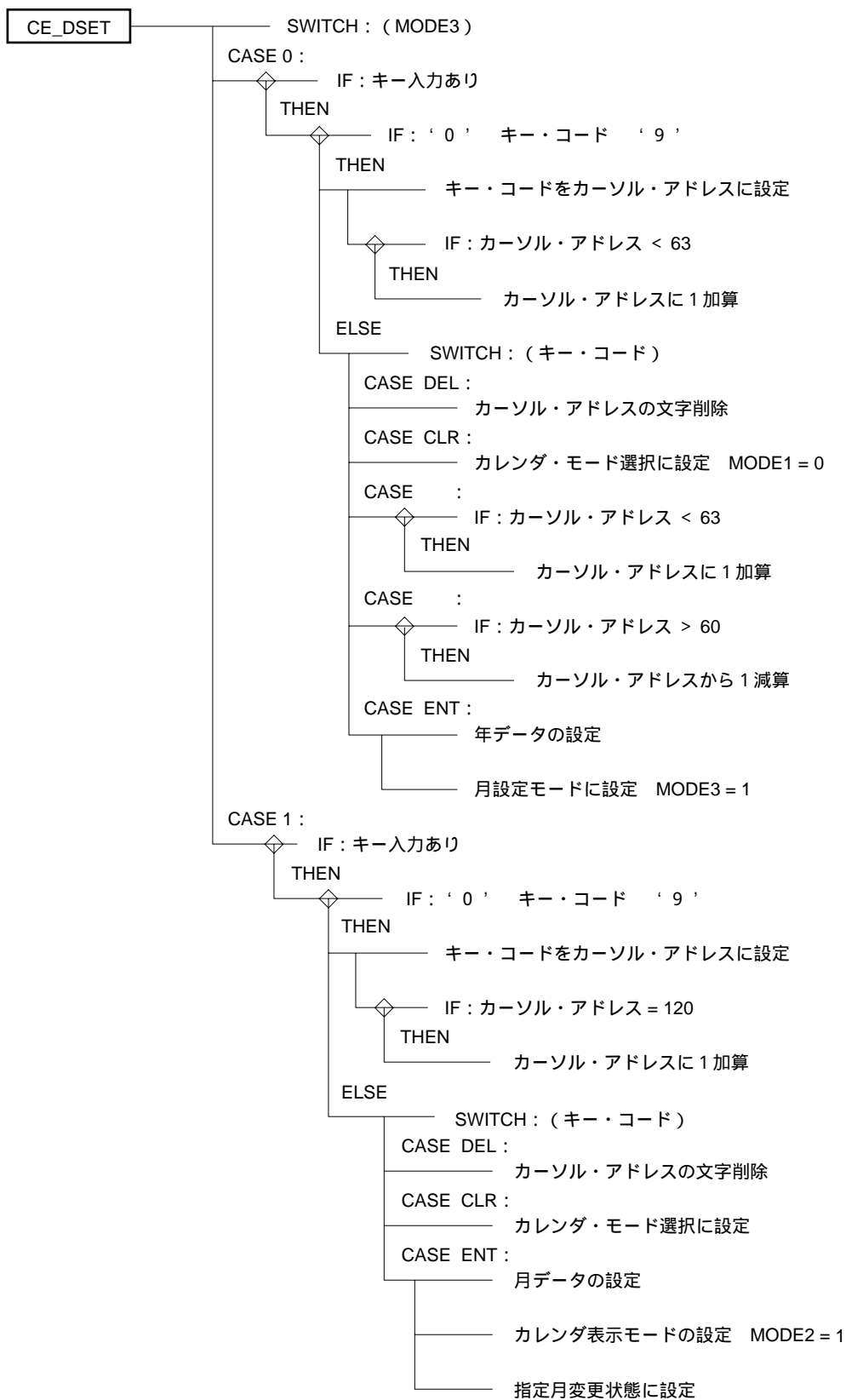
カレンダー・モード

・指示月表示



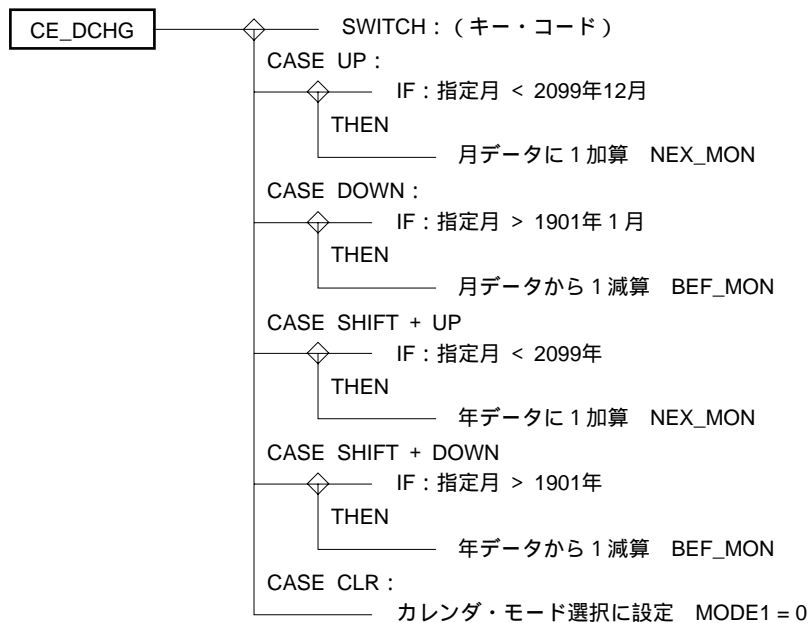
カレンダー・モード

・指定月表示



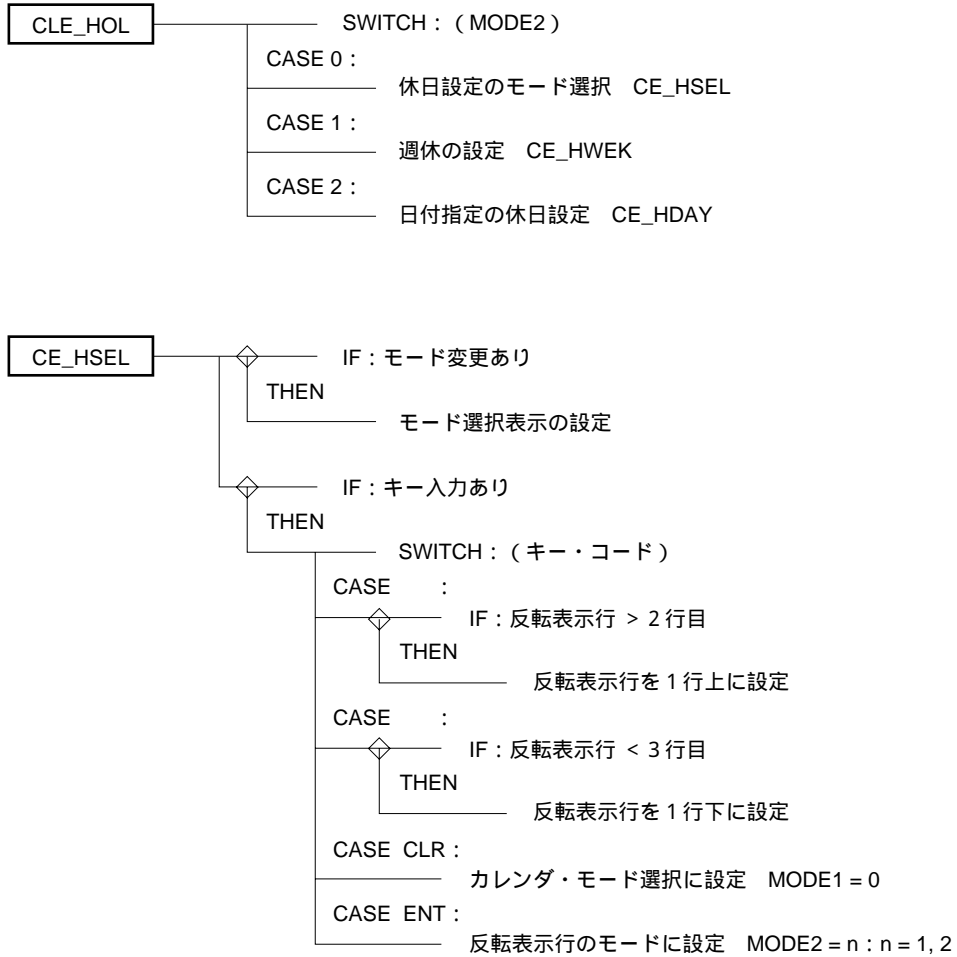
カレンダー・モード

・指定月表示



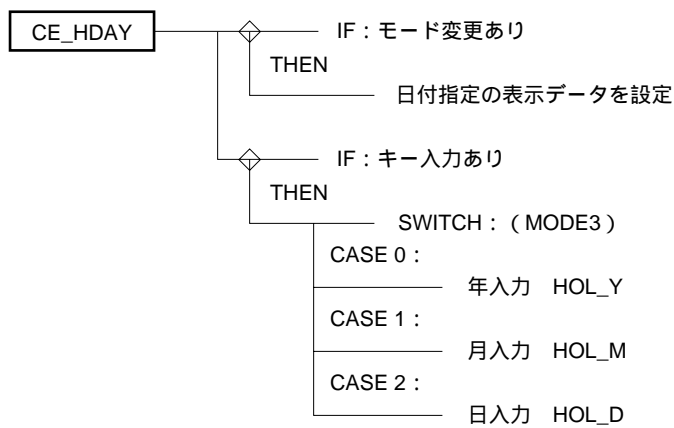
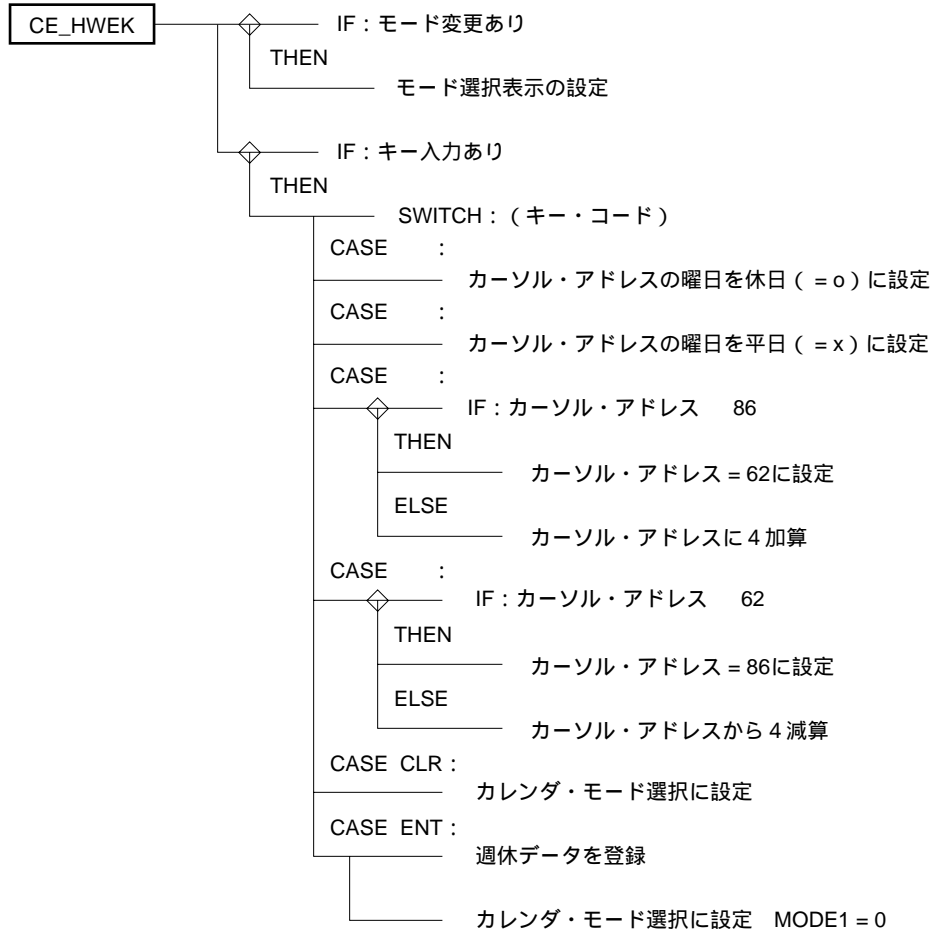
カレンダー・モード

・休日設定



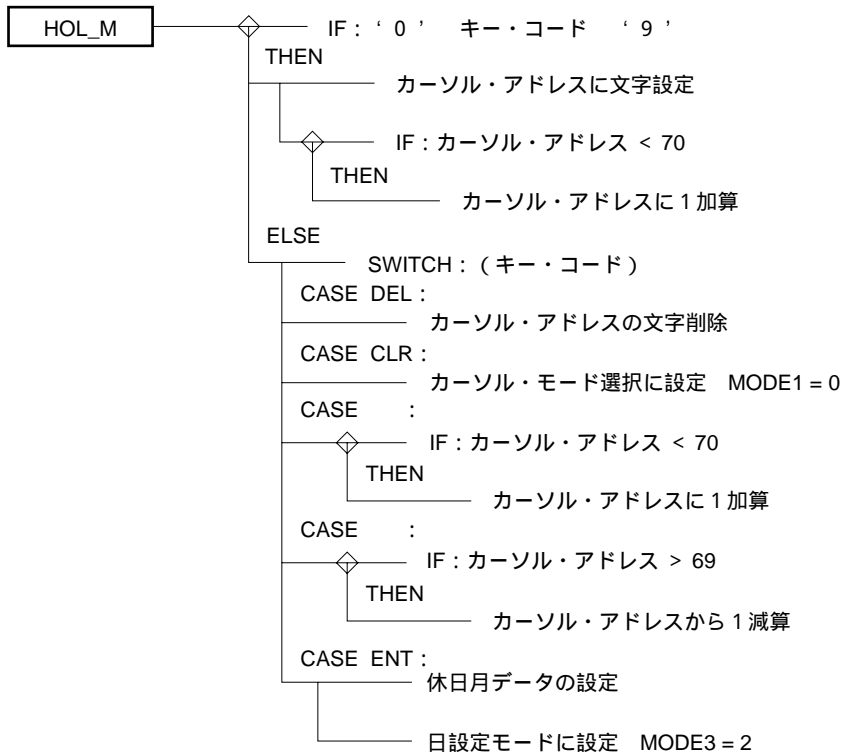
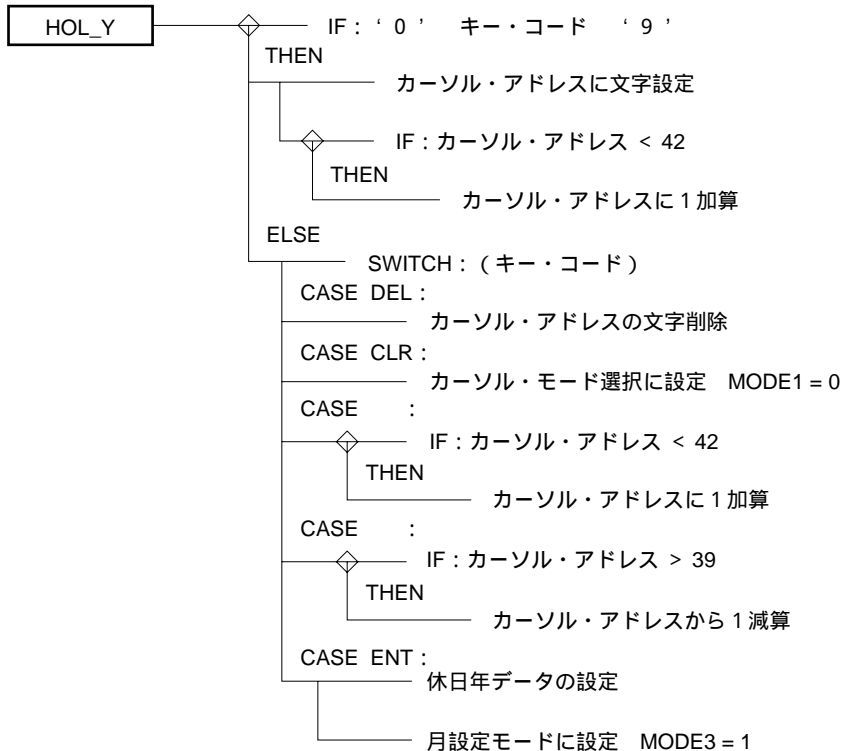
カレンダー・モード

・休日設定



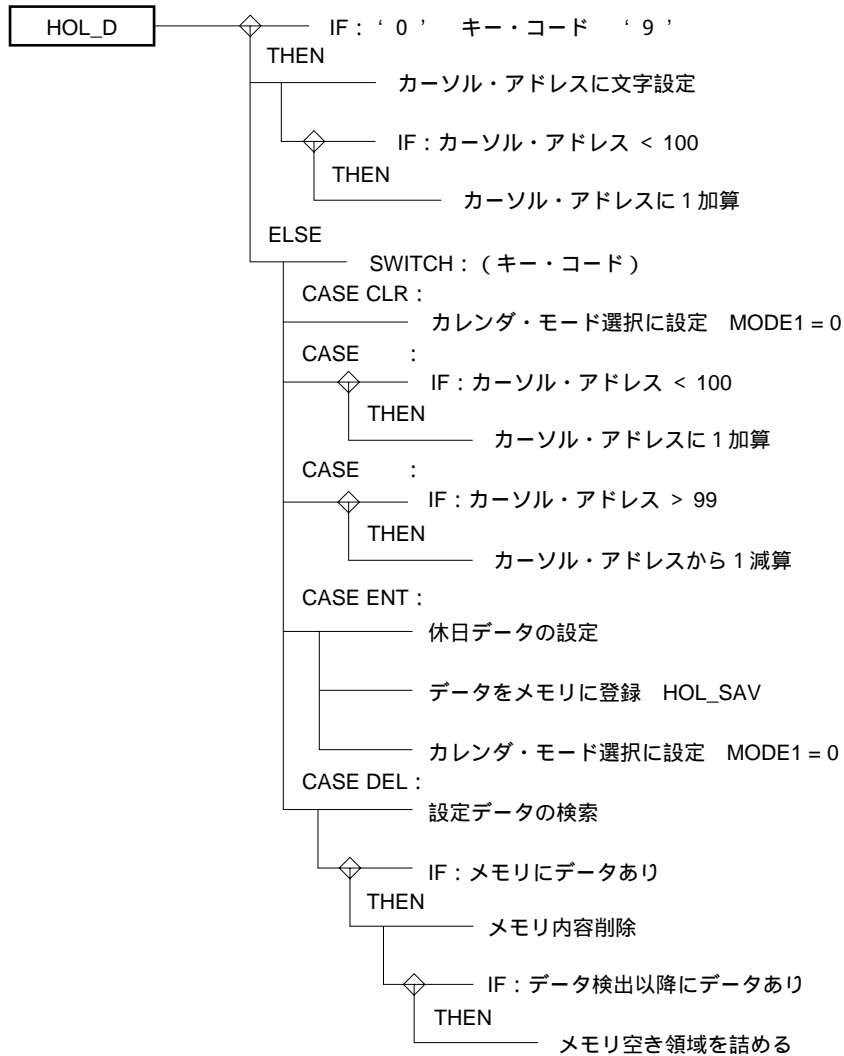
カレンダー・モード

・休日設定



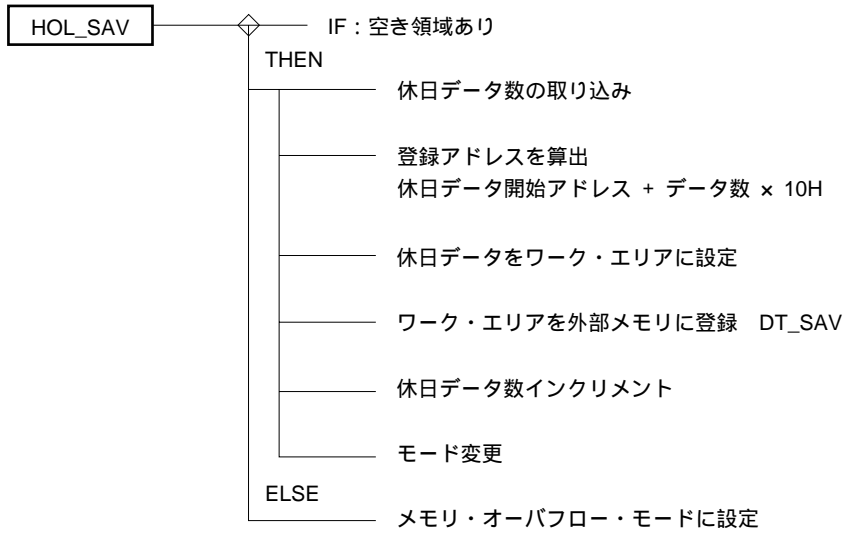
カレンダー・モード

・休日設定

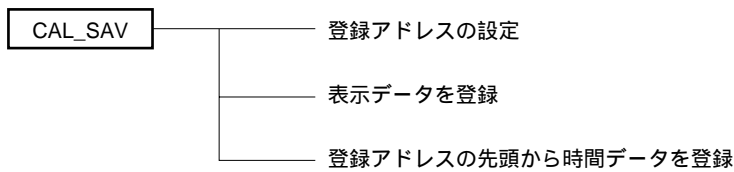


カレンダー・モード

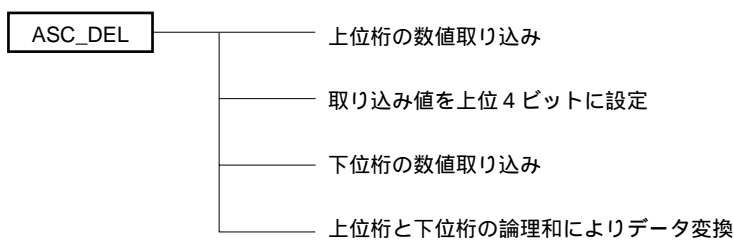
・休日データ登録



・修正モードのデータ登録

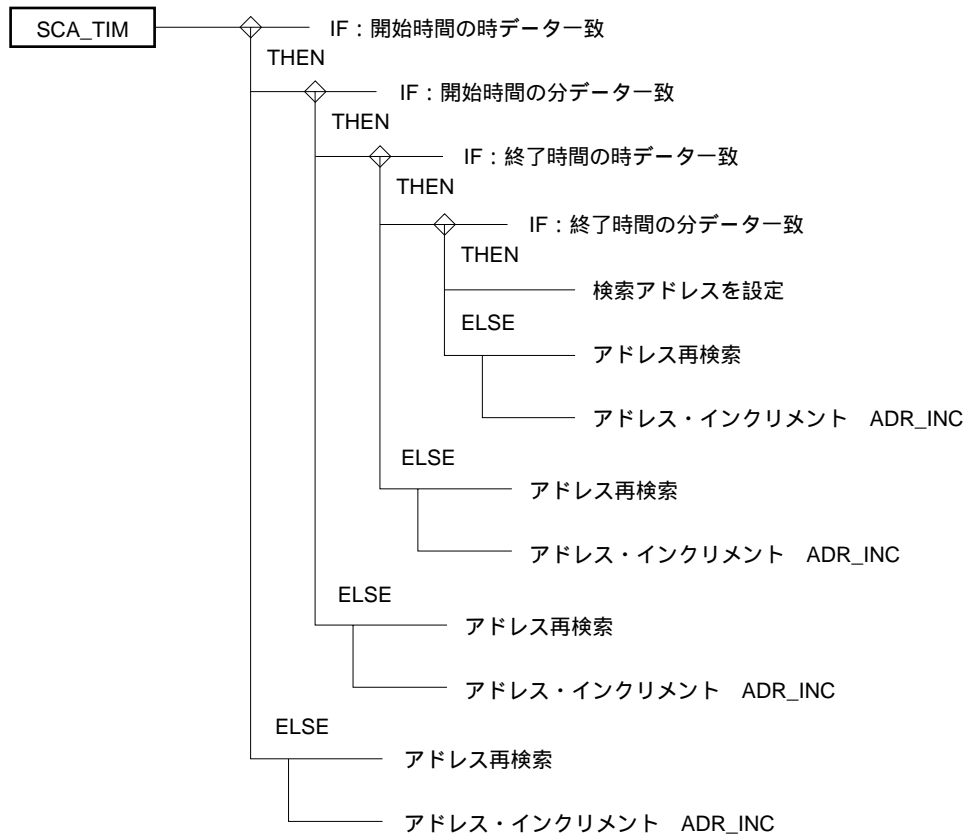


・表示コードのデータ変換



カレンダー・モード

・スケジュール時間比較

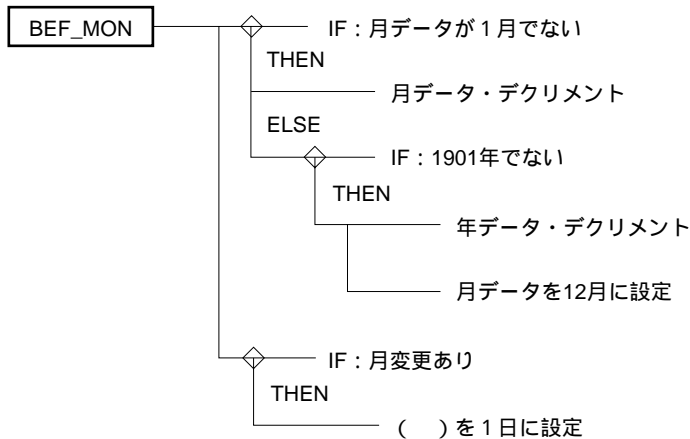


・スケジュール・タイトル表示

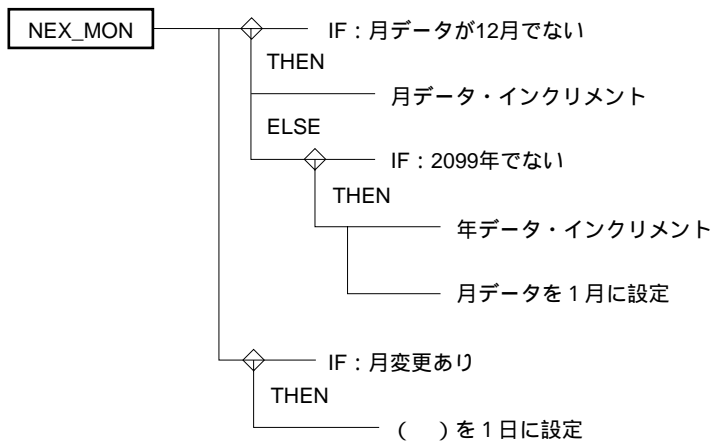


カレンダー・モード

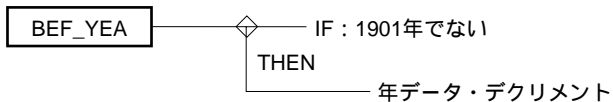
・月変更 (- 1 箇月)



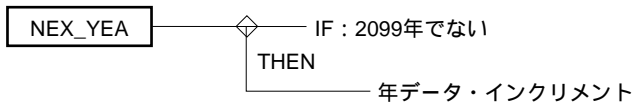
・月変更 (+ 1 箇月)



・年変更 (- 1 年)

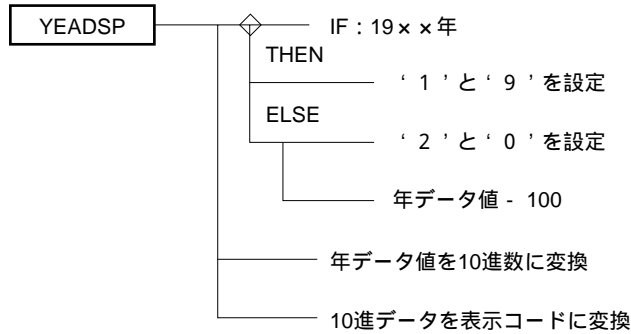


・年変更 (+ 1 年)

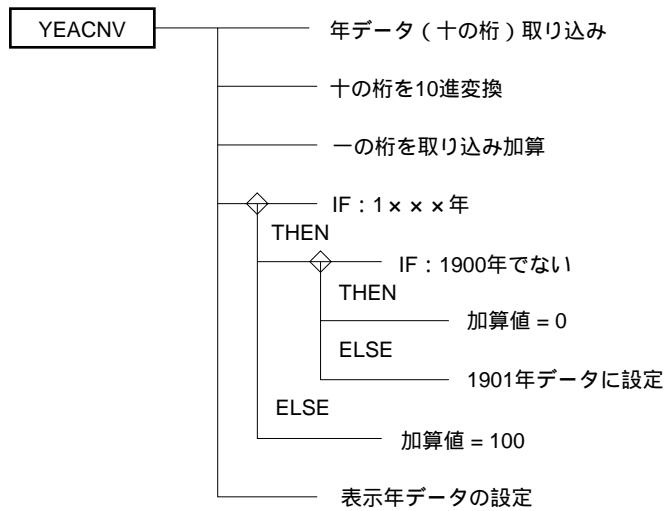


カレンダー・モード

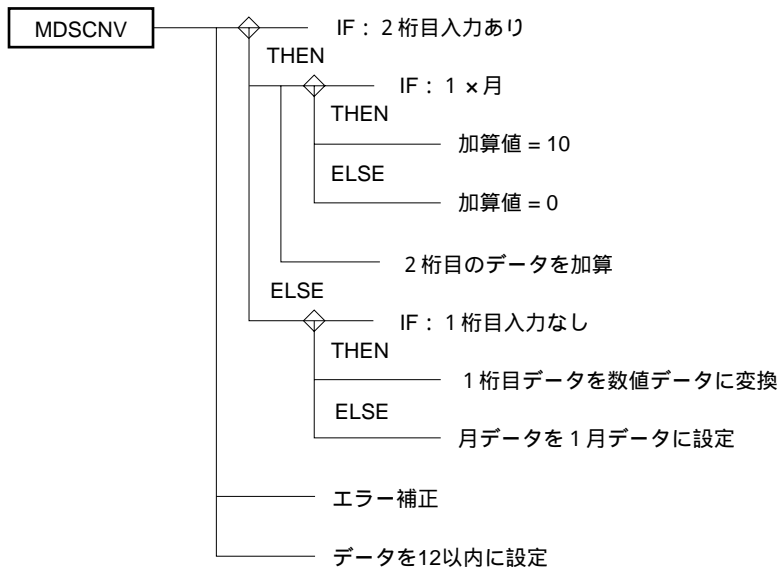
・年表示



・年表示の数値変換

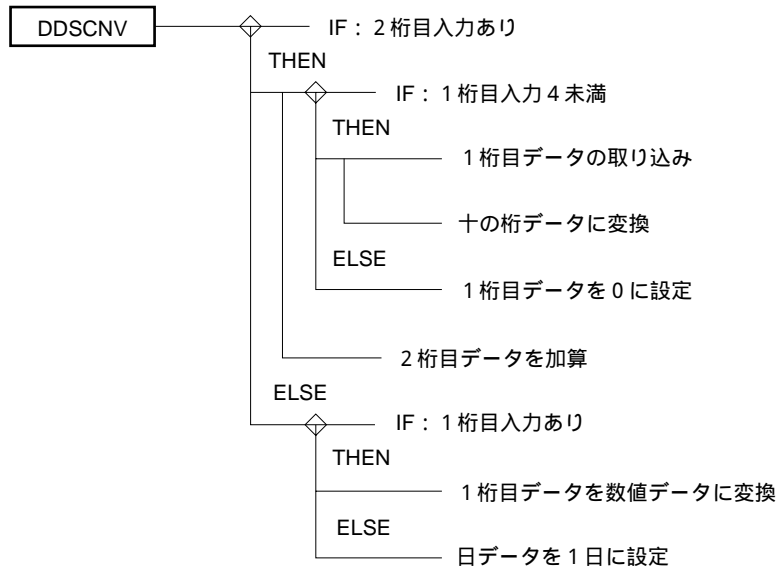


・月表示の数値変換

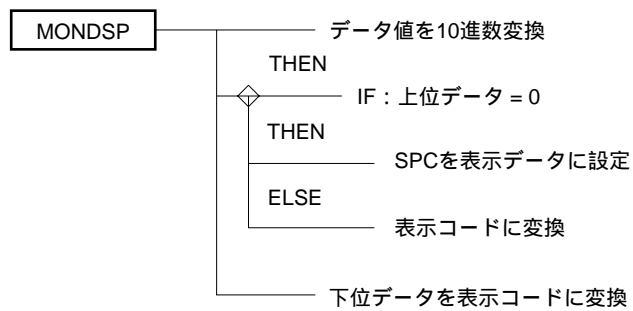


カレンダー・モード

・日表示の数値変換

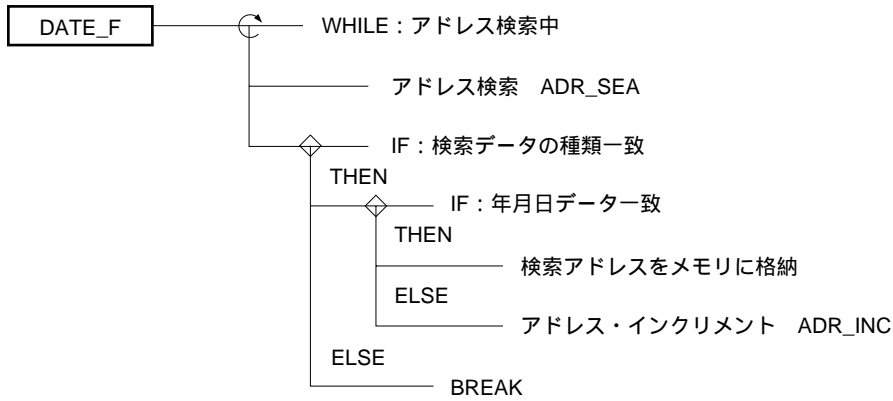


・月日表示

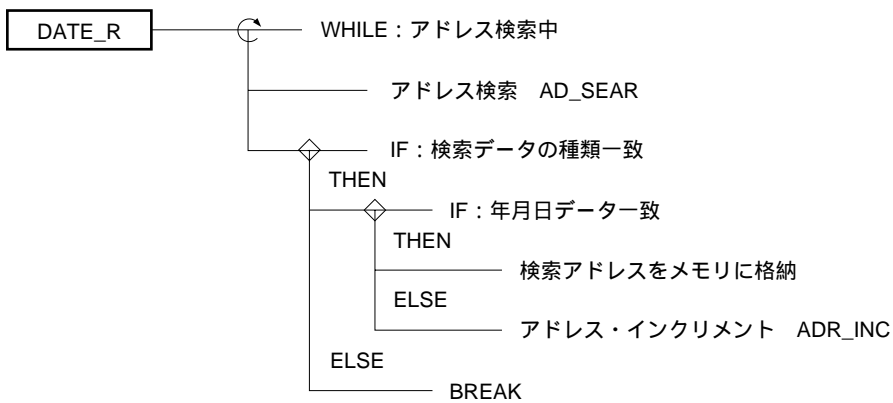


カレンダー・モード

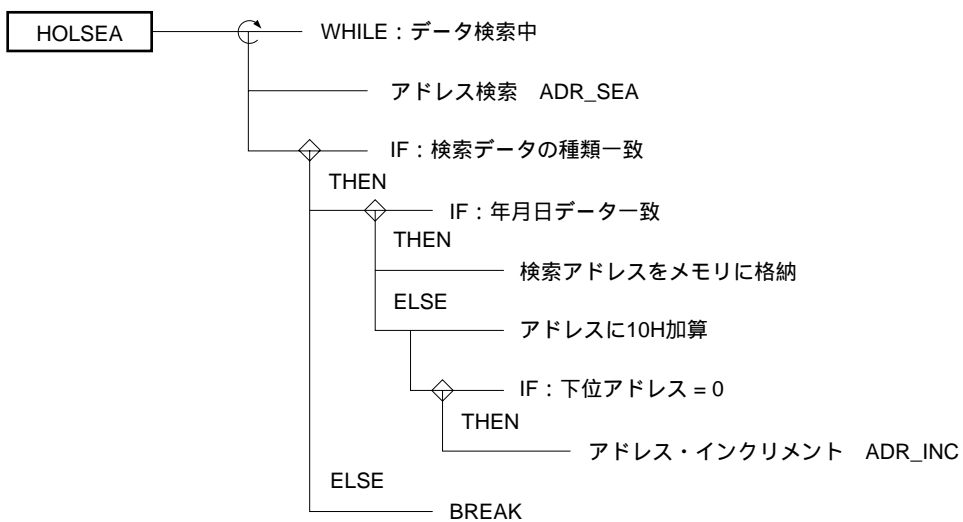
・日付検索（前進）



・日付検索（後退）



・休日検索



4.6.11 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
EXTRN  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, MEMPAG, MEMBNK, BACPAG, BACBNK
EXTRN  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
EXTRN  VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DT1LSET, DT1PSET, LCD_C, LCD_D, DT_SAV, STA_SEA, ASC_DEL
EXTRN  SAVADR, DTCOMP, DT_DEL, DEL_1L, BANKSET, ADR_SEA, AD_SEAR, ADR_INC, ADR_DEC, DT_SET, NEX_ADR, BEF_ADR
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF
EXTBIT  IV_CHGF, DS_ERA, CR_DSPP, ADENDF, ADSEAF, CMPERF, LCDCHGF

PUBLIC  S_CALE, YEADSP, MONDSP, CAL_LEA, CAL_NOR, YDSCNV, MDSCNV, DDSCNV
PUBLIC  CE_YEAD, CE_MOND, CE_DAYD, ALMON, ALMOFF

$      INCLUDE (LCDCMD. INC)
$      INCLUDE (LCDT_EX. INC)

$LIST

DSEG      SADDR
PAR_ADR:DS 1                ;カレンダー( )表示アドレス
CE_YEAD:DS 1                ;カレンダー表示年データ
CE_MOND:DS 1                ;カレンダー表示月データ
CE_DAYD:DS 1                ;カレンダー表示日データ
CE_TIME:DS 4                ;カレンダー検索時間データ
CE_WEKD:DS 1                ;カレンダー曜日データ
WEKHOLD:DS 1                ;週休データ
MON_END:DS 1                ;指定月最終日格納
STADAY: DS 1                ;指定月1日の曜日格納
HOL_VOL:DS 1                ;休日表示データ数

BSEG
CLDCGF DBIT                ;カレンダー月表示変更フラグ
PAR_1F DBIT                ; ( ) 位置の1日設定フラグ
PAR_SET DBIT               ; ( ) 位置設定フラグ
TIM_OKF DBIT               ;スケジュール時間正常判断フラグ

STAWEK EQU 2                ;1901年1月1日=火曜日
ALMON EQU 00H              ;アラームONデータ
ALMOFF EQU 0FFH            ;アラームOFFデータ

;*****
;      カレンダー処理
;*****
CALSEG      CSEG
S_CALE:
  switch(MODE1)
  case 0:
    CALL  ICLE_SEL
    break
  case 1:
    CALL  ICLE_CUR
    break
  case 2:
    CALL  ICLE_DSP
    break
  case 3:
    CALL  ICLE_HOL
  ends
  RET

;*****
;      カレンダー選択処理
;*****

```

```

CLE_SEL:
  if_bit(MD_CG1F)
    CLR1 MD_CG1F
    HL=#LCDDS ;カレンダー・モード選択表示の設定
    DE=#CALEND ;LCD 1行目のデータを設定
    CALL !DTILSET

    DE=#TOD_D ;LCD 2行目のデータを設定
    CALL !DTILSET

    DE=#MONTH_D ;LCD 3行目のデータを設定
    CALL !DTILSET

    DE=#SETHOLI ;LCD 4行目のデータを設定
    CALL !DTILSET

    DE=#CLR1LD ;LCD 5行目のデータを設定
    CALL !DTILSET

    DE=#CLR1LD ;LCD 6行目のデータを設定
    CALL !DTILSET

    DE=#CLR1LD ;LCD 7行目のデータを設定
    CALL !DTILSET

    DE=#CLR1LD ;LCD 8行目のデータを設定
    CALL !DTILSET

    MODE2=#0
    MODE3=#0
    MODE4=#0
    SET1 MD_CG2F
    SET1 MD_CG3F
    SET1 MD_CG4F
    CLR1 CR_DSPF
    SET1 IV_CHGF ;反転表示行を2行目に設定
    INV_DS=#2
    SET1 LCDCHGF
  endif

  if_bit(KY_SIGF)
    CLR1 KY_SIGF
    switch(KEY_CD) ;キー・コード分岐
      case CUR_UP:
        if(INV_DS)#2
          INV_DS--
          SET1 IV_CHGF ;反転表示行変更状態に設定
        endif
        break
      case CUR_DN: ;反転表示行を下に移動
        if(INV_DS)<#4
          INV_DS++
          SET1 IV_CHGF ;反転表示行変更状態に設定
        endif
        break
      case CLEAR: ;前面モードに変更
        MODE0=#0
        SET1 MD_CG0F
        break
      case ENTER: ;反転表示行のモード実行
        A=INV_DS
        A--
        MODE1=A
        SET1 MD_CG1F ;モード変更状態に設定
    ends
  endif
  RET

;*****
; カレンダー現日付表示処理
;*****
CLE_CUR:
  if_bit(MD_CG1F) ;モード変更
  CLR1 MD_CG1F
  INV_DS=#0 ;反転表示削除

```

```

SETI  IV_CHGF
CE_YEAD=YEAD (A)           ;現在日付をカレンダー表示月に設定
CE_MOND=MOND (A)
CE_DAYD=DAYD (A)
SETI  PAR_SET
SETI  LCDCHGF
SETI  CLDCGF

else
  switch(MODE2)
  case 0:
    if_bit(CLDCGF)         ;表示月変更あり
      CLRI  CLDCGF
      CALL  !LCD_DSC
      CALL  !CLDDSP       ;カレンダー月表示
      CALL  !PARDSP       ; ( ) 表示
      CALL  !SCAMAX       ;スケジュール・マーク表示
      CALL  !W_HDSP       ;週休表示
      CALL  !HOLDSP       ;指定休日表示
      SETI  LCDCHGF
      CLRI  CR_DSPP
    endif
    CALL  !CE_CDSP       ;カレンダー表示
    HL=#LCDDS           ;表示アドレスの設定
    B=PAR_ADR (A)
    [HL+B]=#PAR_LE (A)  ; ( データの設定
    A=B
    A+=#4
    B=A
    [HL+B]=#PAR_RI (A)  ; ) データの設定
    break
  case 1:
    CALL  !CE_CSC       ;スケジュール・モードに設定
  ends
endif
RET

;*****
; カレンダー表示処理
;*****
CLDDSP:
  B=#STAVEK           ;1901年1月1日の曜日設定
  A=CE_YEAD           ; ( (指定年-1901) + (指定年-1901) / 4
  A--                ;指定年-1901+開始曜日
  B+=A
  A>>=1              ; (指定年-1901) / 4を加算
  A>>=1
  B+=A
  A=CE_YEAD           ; ( (指定年-1901) + (指定年-1901) / 4
  A&=#00000011B      ;指定月の曜日移動日数の設定
  if_bit(Z)
    C=CE_MOND (A)
    C--
    HL=#CAL_LEA       ;うるう年の月データ参照
    MON_END=[HL+C] (A) ;指定月の最終日を設定
    HL=#CALE_T
    A=[HL+C]
    A>>=1              ;うるう年：テーブル・データの上4ビットを取り込み
    A>>=1
    A>>=1
    A>>=1
  else
    C=CE_MOND (A)
    C--
    HL=#CAL_NOR       ;平年の月データ参照
    MON_END=[HL+C] (A) ;指定月の最終日を設定
    HL=#CALE_T
    A=[HL+C]
  endif
  A&=#00001111B      ;平年：テーブル・データの下4ビットを取り込み
  A+=B                ;指定月の曜日移動日数を加算
  X=#0
  A<->X
  C=#7
  AX/=C               ;1週間日数(7日間)で除算

```



```

STADAY=C (A) ;除算結果の余り=指定月1日の曜日
CALL !LCDCLR ;LCD表示エリアのクリア
HL=#LCDDS+12 ;日付設定先頭アドレス
X=CE_YEAD (A) ;年データの取り込み
CALL !YEADSP ;年データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=CE_MOND (A) ;月データの取り込み
CALL !MONDSP ;月データ表示
DE=#WEEK_D ;曜日データ・テーブルの設定
HL=#LCDDS+30 ;表示開始アドレスの設定
CALL !DTILSET
A=#6 ;カレンダー表示データの転送開始アドレスの作成
A=STADAY
A<<=1 ;1データ=2バイト
X=A ;アドレス修正データに変換(データ×2倍)
A=#0
AX+=#CALDSP ;カレンダー表示データの転送開始アドレスを設定
DE=AX
HL=#LCDDS+62 ;カレンダー表示開始アドレス(3行目)
for(i=#0;i<#6;i++) ;3-8行(6行分)のデータを転送
    CALL !CALDSET ;1行分のデータを転送
    HL++ ;次の行の開始アドレスを設定
    HL++
next
if(MON_END!=#31)
    A=STADAY ;1日の開始アドレスを設定
    A<<=1 ;1日分の表示4バイト
    A<<=1
    X=A ;表示開始アドレス=62+4×n
    switch(MON_END)
    case 28:
        A=#0
        AX+=#LCDDS+62+120 ;29日以降表示クリア
        break
    case 29:
        A=#0
        AX+=#LCDDS+62+124 ;30日以降表示クリア
        break
    case 30:
        A=#0
        AX+=#LCDDS+62+128 ;31日以降表示クリア
    ends
    HL=AX
    while(HL<#LCDDS+216) (AX) ;31日格納最大アドレス=LCDDS+217
        [HL]=#SPC (A)
        HL++
    endw
endif
RET

;*****
; ( ) アドレスの設定
;*****
PARDSP:
    if_bit(PAR_SET) ; ( ) アドレスを指定日に変更
        CLR1 PAR_SET
        A=CE_DAYD ;指定日の取り込み
        CALL !DEF_ADR
        A+#60 ; ( ) 開始アドレスを加算
        PAR_ADR=A ; ( ) アドレスの設定
    elseif_bit(PAR_1F) ; ( ) を1日に設定
        CLR1 PAR_1F
        A=STADAY ;曜日データからアドレスを算出(×4)
        A<<=1
        A<<=1
        A+#60 ; ( ) 開始アドレスを加算
        PAR_ADR=A ; ( ) アドレスの設定
        CE_DAYD=#1
    endif
RET

;*****
; スケジュール・マークの設定
;*****

```

```

SCAMAK:
B=#SCA_DAT                ;スケジュール・データの設定
M_BANK=#0                 ;メモリ先頭アドレスに設定
CALL !BANKSET
H=#ADRSTA
L=#0
while(M_BANK<#4)          ;データ検索中
  CALL !SCA_CMP           ;スケジュール・データの比較
  if_bit(ADSEAF)         ;スケジュール・データあり
  A=[HL+3]               ;日付データの取り込み
  CALL !DEF_ADR          ;表示アドレスの算出
  A+=#61
  X=A
  A=#0
  AX+=#LCDDS             ;スケジュール・マークのアドレスを設定
  DE=AX
  [DE]=#MUL_SIG(A)      ;*データを設定
  CALL !ADR_INC
  if_bit(!CY)
    if(M_BANK<#3)       ;バンク0-バンク2
      CALL !BANKSET
    endif
  endif
  elseif_bit(!CY)
    if(M_BANK<#3)       ;バンク0-バンク2
      CALL !BANKSET
    endif
  endif
endif
endw
RET

;*****
;  週休表示の設定
;*****
W_HDSP:
if_bit(WEKHOLD.0)        ;日曜日=休日
HL=#1000H+480+2         ;日曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+3         ;日曜日下位桁に反転データ設定
CALL !W_HOLI
endif

if_bit(WEKHOLD.1)        ;月曜日=休日
HL=#1000H+480+6         ;月曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+7         ;月曜日下位桁に反転データ設定
CALL !W_HOLI
endif

if_bit(WEKHOLD.2)        ;火曜日=休日
HL=#1000H+480+10        ;火曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+11        ;火曜日下位桁に反転データ設定
CALL !W_HOLI
endif

if_bit(WEKHOLD.3)        ;水曜日=休日
HL=#1000H+480+14        ;水曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+15        ;水曜日下位桁に反転データ設定
CALL !W_HOLI
endif

if_bit(WEKHOLD.4)        ;木曜日=休日
HL=#1000H+480+18        ;木曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+19        ;木曜日下位桁に反転データ設定
CALL !W_HOLI
endif

if_bit(WEKHOLD.5)        ;金曜日=休日
HL=#1000H+480+22        ;金曜日上位桁に反転データ設定
CALL !W_HOLI
HL=#1000H+480+23        ;金曜日下位桁に反転データ設定
CALL !W_HOLI

```

```

endif

if_bit(WEKHOLD.6)                ;土曜日 = 休日
    HL=#1000H+480+26            ;土曜日上位桁に反転データ設定
    CALL !W_HOLI
    HL=#1000H+480+27            ;土曜日下位桁に反転データ設定
    CALL !W_HOLI
endif
RET

;*****
; 指定休日表示の設定
;*****
HOLDSP:
    H=#ADRSTA                    ;開始アドレスの設定
    L=#00H
    M_BANK=#0
    CALL !BANKSET
    while(forever)
        CALL !HOL_SET            ;指定休日のデータ検索
        if_bit(ADSEAF)          ;検索データあり
            PUSH HL
            A=[HL+3]            ;日付データの取り込み
            CALL !DEFAD_H
            CALL !HOL_MAK        ;休日マークの設定
            POP HL
            CALL !HADRINC
            if_bit(ICY)
                break
            endif
        else
            break                ;データ検索終了
        endif
    endw
    RET

;*****
; 週休表示設定処理
;*****
W_HOLI:
    A=(CSRDIR OR 00000011B)      ;カーソル・シフト方向を下に設定
    !LCD_C=A
    !LCD_C=#CSRW (A)            ;LCDカーソル位置の設定
    !LCD_D=L (A)                ;HLレジスタによりアドレス設定
    !LCD_D=H (A)
    !LCD_C=#WRITE (A)           ;データ書き込み
    A=#0FFH                      ;8ドット表示データを設定
    for(i=#0;i<#8*6;i++)        ;3-8行(8ドット×6行)にデータ書き込み
        !LCD_D=A
    next
    RET

;*****
; 指定休日の表示設定処理
;*****
HOL_MAX:
    A=(CSRDIR OR 00000011B)      ;カーソル・シフト方向を下に設定
    !LCD_C=A
    !LCD_C=#CSRW (A)            ;LCDカーソル位置の設定
    !LCD_D=L (A)                ;HLレジスタによりアドレス設定
    !LCD_D=H (A)
    !LCD_C=#WRITE (A)           ;データ書き込み
    A=#0FFH                      ;8ドット表示データを設定
    for(i=#0;i<#8;i++)          ;1文字分(8ドット)のデータ書き込み
        !LCD_D=A
    next
    HL++                          ;日付下位アドレスの設定
    !LCD_C=#CSRW (A)            ;LCDカーソル位置の設定
    !LCD_D=L (A)                ;HLレジスタによりアドレス設定
    !LCD_D=H (A)
    !LCD_C=#WRITE (A)           ;データ書き込み
    A=#0FFH                      ;8ドット表示データを設定
    for(i=#0;i<#8;i++)          ;1文字分(8ドット)のデータ書き込み
        !LCD_D=A

```

```

next
RET

;*****
; スケジュール日付比較処理
;*****
SCA_CMP:
  CLR1  ADSEAF
  while_bit(!ADSEAF)
    CALL  !ADR_SEA
    if_bit(CY)
      if([HL]==B) (A)
        if(CE_YEAD==[HL+1]) (A)
          if(CE_MOND==[HL+2]) (A)
            SET1  ADSEAF
            MEMPAG=H (A)
          ;アドレス決定
          ;メモリ・ページの格納
        else
          CALL  !ADR_INC
          if_bit(!CY)
            break
          endif
        endif
      else
        CALL  !ADR_INC
        if_bit(!CY)
          break
        endif
      endif
    else
      CALL  !ADR_INC
      if_bit(!CY)
        break
      endif
    endif
  endwhile
  RET

;*****
; 指定休日検索処理
;*****
HOL_SET:
  CLR1  ADSEAF
  while([HL]<=#HOL_DAT) (A)
    if([HL]==#HOL_DAT) (A)
      if(CE_YEAD==[HL+1]) (A)
        if(CE_MOND==[HL+2]) (A)
          SET1  ADSEAF
          break
        ;検索データのアドレス決定
      else
        CALL  !HADRINC
        if_bit(!CY)
          break
        endif
      endif
    else
      CALL  !HADRINC
      if_bit(!CY)
        break
      endif
    endif
  endwhile
  CALL  !ADR_INC
  if_bit(!CY)
    if(M_BANK<=#3)
      CALL  !BANKSET
      ;バンク変更
    else
      break
    endif
  endif
  ;次アドレスに設定

```

```

        endif
    endif
endw
RET

;*****
;  週休データ・アドレス検出処理
;*****
HADRINC:
    if(HL<#0BFF0H) (AX)                ;最終アドレス・データでない
        AX=HL                          ;次アドレスに設定
        AX+="#10H
        HL=AX
        SETI  CY
    else
        if(M_BANK<#3)                  ;最終バンクでない
            M_BANK++
            CALL  IBANKSET              ;バンク変更
            H=#ADRSTA                   ;開始アドレスに設定
            L=#00H
            SETI  CY
        else
            CLRI  CY                    ;データ検索終了
        endif
    endif
    RET

;*****
;  カレンダー日付アドレス算出処理
;*****
DEF_ADR:
    A+=STADAY                          ;1日の曜日データを加算
    A--
    X=A
    A=#0
    C=#7
    AX/=C                                ;指定日の週を算出
    if(X==#0) (A)                       ;Zフラグの設定
    endif
    A=C                                  ;曜日データからアドレスを算出(×4)
    ROL  A,1
    ROL  A,1
    while_bit(1Z)
        A+="#30                          ;指定日の週×30バイトを加算
        X--
    endwhile
    RET

;*****
;  カレンダー休日アドレス算出処理
;*****
DEFAD_H:
    A+=STADAY                          ;1日の曜日データを加算
    A--
    X=A
    A=#0
    C=#7
    AX/=C                                ;指定日の週を算出
    if(X==#0) (A)                       ;Zフラグの設定
    endif
    B=X (A)                              ;曜日データからアドレスを算出(×4)
    A=C
    ROL  A,1 ;A<<=1
    ROL  A,1 ;A<<=1
    X=A
    A=#0
    while_bit(1Z)
        AX+="#240                        ;指定日の週×240バイトを加算
        B--
    endwhile
    AX+="#1000H+482                      ;日付表示先頭アドレス
    HL=AX
    RET

;*****

```

```

; カレンダ・データ1ライン設定処理
;*****
CALDSET:
    for(j=#0;j<#7;j++)
        [HL]=[DE] (A)
        HL++
        DE++
        [HL]=[DE] (A)
        HL++
        HL++
        HL++
        DE++
    next
    RET

;*****
; カレンダ表示用キー入力データ
;*****
CE_CDSP:
    IF_BIT(KY_SIGF)
        CLR1 KY_SIGF
        switch(KEY_CD)
            case CUR_UP:
                if(PAR_ADR>#89)
                    A=PAR_ADR
                    A-=#27
                    B=A
                    HL=#LCDD5
                    if([HL+B]==#SPC) (A)
                        CALL !BEF_MON
                    else
                        PAR_ADR-=#30
                        CE_DAYD-=#7
                    endif
                else
                    CALL !BEF_MON
                endif
                SET1 CLDCGF
                break
            case CUR_DN:
                if(PAR_ADR<#210)
                    A=PAR_ADR
                    A+=#33
                    B=A
                    HL=#LCDD5
                    if([HL+B]==#SPC) (A)
                        CALL !NEX_MON
                    else
                        PAR_ADR+=#30
                        CE_DAYD+=#7
                    endif
                else
                    CALL !NEX_MON
                endif
                SET1 CLDCGF
                break
            case CUR_R1:
                X=PAR_ADR (A)
                A=#0
                C=#30
                AX/=C
                if(C==#24) (A)
                    B=#9
                else
                    B=#7
                endif
                A=PAR_ADR
                B+=A
                HL=#LCDD5
                if([HL+B]==#SPC) (A)
                    CALL !NEX_MON
                else
                    A=B
                    A-=#3
                    PAR_ADR=A
                endif
        endswitch
; キー・コード分岐
; ( ) が2行目以降にあり
; 1行上の文字取り込み
; ( ) に日付なし
; 前の月に設定
; 1行上の行に設定
; ( ) が7行目以前にあり
; 1行下の文字取り込み
; ( ) に日付なし
; 次の月に設定
; 1行下の行に設定
; 次の月に設定
; ( ) 位置の取り込み
; 桁数の判別 (除算結果の余り)
; 24桁目 (土曜日)
; 次の行の検索 (日曜日)
; 右の桁の検索
; ( ) に日付なし
; 次の月に設定
; 次の日に ( ) を設定

```

```

        CE_DAYD+=#1
    endif
    SET1 CLDCGF
    break
case CUR_LE:
    X=PAR_ADR (A)           ; ( ) 位置の取り込み
    A=#0
    C=#30                   ; 桁数の判別 (除算結果の余り)
    AX/=C
    B=PAR_ADR (A)
    if(C=#0) (A)           ; 1 桁目 (日曜日)
        A=#3               ; 前の行の検索 (土曜日)
    else
        A=#1               ; 左の桁の検索
    endif
    B-=A
    HL=#LCDDS
    if([HL+B]==#SPC) (A)   ; ( ) に日付なし
        CALL !BEF_MON     ; 前の月に設定
    else
        A=B                 ; 次の日に ( ) を設定
        A-=#3
        PAR_ADR=A
        CE_DAYD-=#1
    endif
    if(PAR_ADR<#60)       ; ( ) が日付行でない
        CALL !BEF_MON     ; 前の月に設定
    endif
    SET1 CLDCGF
    break
case UP:
    CALL !NEX_MON
    SET1 CLDCGF
    break
case DOWN:
    CALL !BEF_MON
    SET1 CLDCGF
    break
case S_UP:
    if(CE_YEAD<#199)      ; 2009年以前
        CE_YEAD++         ; 次の年に設定
        SET1 CLDCGF
        SET1 PAR_IF       ; ( ) を1日に設定
    endif
    break
case S_DOWN:
    if(CE_YEAD>#1)       ; 1901年以降
        CE_YEAD--         ; 前の年に設定
        SET1 CLDCGF
        SET1 PAR_IF       ; ( ) を1日に設定
    endif
    break
case CLEAR:
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:
    MODE2=#1
    SET1 MD_CG3F
    SET1 MD_CG4F
    FOR_BNK=#0
    FOR_ADR=#ADRSTA
    REV_BNK=#0
    REV_ADR=#ADRSTA
ends
ENDIF
RET

;*****
; スケジュール・モード
;*****
CE_CSC:
switch(MODE3)
case 0:
    CALL !SC_SEL

```

```

        break
case 1:
    CALL !SC_REG
    break
case 2:
    CALL !SC_COR
ends
RET

;*****
;   スケジュール選択
;*****
SC_SEL:
    if_bit(MD_CG3F)
        CLR1 MD_CG3F
        CALL !SCA_DSP
        NODE4=#0
        SET1 MD_CG4F
        FOR_ADR=#ADRSTA
        FOR_BNK=#0
        REV_ADR=#ADRSTA
        REV_BNK=#0
    endif

    IF_BIT(MD_CG4F)
        CLR1 MD_CG4F
        SWITCH(NODE4)
        CASE 0:
            FOR(j=#0;j<#5;j++)
                H=FOR_ADR(A)
                L=#00H
                M_BANK=FOR_BNK(A)
                CALL !BANKSET
                B=#SCA_DAT
                CALL !DATE_F
                IF_BIT(ADSEAF)
                    PUSH HL
                    switch(j)
                    case 0:
                        CALL !NEX_ADR
                        CALL !SCA_DSP
                        HL=#LCDDS+62
                        break
                    case 1:
                        HL=#LCDDS+92
                        break
                    case 2:
                        HL=#LCDDS+122
                        break
                    case 3:
                        HL=#LCDDS+152
                        break
                    case 4:
                        HL=#LCDDS+182
                        break
                    ends
                POP DE
                E=#16+41
                B=#5
                CALL !DT_SET
                HL++
                [HL]=#SUB_SIG(A)
                HL++
                E=#16+71
                B=#5
                CALL !DT_SET
                HL++
                HL++
                E=#16+131
                B=#14
                CALL !DT_SET
                H=D(A)
                CALL !ADR_INC
                FOR_ADR=H(A)
                FOR_BNK=M_BANK(A)
            ; 1 ページ表示数のカウント
            ; 現在のメモリ・アドレスを設定
            ; 転送元アドレスの設定
            ; 転送先アドレスの設定
            ; 3 行目に設定
            ; 4 行目に設定
            ; 5 行目に設定
            ; 6 行目に設定
            ; 7 行目に設定
            ; Hレジスタの値を取り込み
            ; 開始時間の表示 2 行目 1 1 桁から
            ; データ転送文字数
            ; - 表示
            ; 終了時間の表示 3 行目 1 1 桁から
            ; データ転送文字数
            ; 用件の表示 5 行目 1 1 桁から
            ; データ転送文字数

```



```

        elseif_bit(ADENDF)          ;検索データなし
        break
    ENDF
NEXT
break
CASE 1:
FOR(j=#0;j<#5;j++)                ;1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)             ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    B=#SCA_DAT
    CALL !DATE_R
    IF_BIT(ADSEAF)
        PUSH HL                    ;転送元アドレスの設定
        switch(j)                  ;転送先アドレスの設定
        case 0:                    ;7 行目に設定
            CALL !BBF_ADR
            CALL !SCA_DSP
            HL=#LCDDS+182
            break
        case 1:                    ;6 行目に設定
            HL=#LCDDS+152
            break
        case 2:                    ;5 行目に設定
            HL=#LCDDS+122
            break
        case 3:                    ;4 行目に設定
            HL=#LCDDS+92
            break
        case 4:                    ;3 行目に設定
            HL=#LCDDS+62
            break
        ends
        POP DE                      ;Hレジスタの値を取り込み
        E=#16+41                    ;開始時間の表示 2行目11桁から
        B=#5                        ;データ転送文字数
        CALL !DT_SET
        HL++
        [HL]=#SUB_SIG (A)           ;-表示
        HL++
        E=#16+71                    ;終了時間の表示 3行目11桁から
        B=#5                        ;データ転送文字数
        CALL !DT_SET
        HL++
        HL++
        E=#16+131                   ;用件の表示 5行目11桁から
        B=#14                       ;データ転送文字数
        CALL !DT_SET
        H=D (A)
        CALL !ADR_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
    elseif_bit(ADENDF)            ;検索データなし
    break
ENDIF
NEXT
ENDS
ENDIF

IF_BIT(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD)                   ;キー・コード分岐
case CUR_UP:
    if(INV_DS>#2)                 ;3 行目以降
        X=INV_DS (A)              ;反転表示行の取り込み
        X--
        A=#30                     ;1 行文字数
        AX*=X
        AX+=#LCDDS+4              ;開始時間の : アドレスの設定
        HL=AX
        repeat
            INV_DS--              ;1 行上の行を参照
            AX=HL
            AX-=#30

```

```

        HL=AX
        until([HL]!="#SPC) (A) ;スペースのときは参照行を1行上に設定
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case CUR_DN: ;反転表示行を下に移動
    if(INV_DS<#8) ;8行目より上の行
        X=INV_DS (A) ;反転表示行の取り込み
        X--
        A=#30 ;1行文字数
        AX*=X
        AX+=#LCDDS+4 ;開始時間の：アドレスの設定
        HL=AX
        repeat
            INV_DS++ ;1行下の行を参照
            AX=HL
            AX+=#30
            HL=AX
        until([HL]!="#SPC) (A) ;スペースのときは参照行を1行下に設定
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case UP:
    MODE4=#0 ;モード変更
    SET1 MD_CG4F
    break
case DOWN:
    MODE4=#1 ;モード変更
    SET1 MD_CG4F
    break
case CLEAR:
    MODE2=#0 ;前面モードに変更
    SET1 MD_CG2F
    SET1 CLDCGF
    break
CASE ENTER: ;反転表示行のモード実行
    if(INV_DS==#2) ;REGISTER選択
        MODE3=#1
        MODE4=#0
        SET1 MD_CG4F ;モード変更状態に設定
    elseif(INV_DS==#8) ;DELETE選択
        M_BANK=#0 ;外部メモリの開始アドレスに設定
        CALL !BANKSET
        MEMPAG=#ADRSTA
        B=#SCA_DAT
        repeat
            H=MEMPAG (A) ;検索アドレスの設定
            L=#0
            M_BANK=MEMBANK (A)
            CALL !BANKSET
            CALL !DATE_F ;データ検索(日付)
            if_bit(ADSEAF) ;データあり
            CALL !DT_DEL ;検索データの削除
        endif
        until_bit(!ADSEAF)
        MODE2=#0
        SET1 CLDCGF
    else ;タイトル選択
        MEMBANK=#0 ;外部メモリの開始アドレスの設定
        MEMPAG=#ADRSTA
        X=INV_DS (A) ;反転表示の行数取り込み
        X--
        A=#30
        AX*=X ;行数から表示アドレスを算出
        AX+=#LCDDS+2
        HL=AX
        X=[HL] (A) ;開始時間の時データ取り込み
        HL++
        A=[HL]
        A<->X
        HL++
        CALL !ASC_DEL ;2バイト表示データを1バイト数値データに変換
        CE_TIME=A ;スケジュール選択データの設定
        HL++ ;' : ' データ・スキップ
        X=[HL] (A) ;開始時間の分データ取り込み

```

```

HL++
A=[HL]
A<->X
HL++
CALL !ASC_DEL ; 2バイト表示データを1バイト数値データに変換
CE_TIME+1=A ; スケジュール選択データの設定
HL++ ; ' ' データ・スキップ
HL++
X=[HL] (A) ; 終了時間の時データ取り込み
HL++
A=[HL]
A<->X
HL++
CALL !ASC_DEL ; 2バイト表示データを1バイト数値データに変換
CE_TIME+2=A ; スケジュール選択データの設定
HL++ ; : データ・スキップ
X=[HL] (A) ; 終了時間の分データ取り込み
HL++
A=[HL]
A<->X
CALL !ASC_DEL ; 2バイト表示データを1バイト数値データに変換
CE_TIME+3=A ; スケジュール選択データの設定
H=NEWPAG (A) ; 検索アドレスの設定
L=#0
B=#SCA_DAT
M_BANK=MEMBNK (A)
CALL !BANKSET
repeat
CALL !DATE_F ; データ検索 (日付)
if_bit(ADSEAF) ; データあり
CALL !SCA_TIM ; データ比較 (時間)
endif
until_bit(ADSEAF)
MODE3=#2 ; スケジュール・データ修正モードに設定
MODE4=#0
SET1 MD_CG4F
endif
ends
ENDIF
RET

;*****
; スケジュール登録モード
;*****
SC_REG:
switch(MODE4)
case 0:
if_bit(MD_CG4F)
CLR1 MD_CG4F
CALL !LCD_DSC
HL=#LCDDS ; 日付設定先順アドレス
X=CE_YEAD (A) ; 年データの取り込み
CALL !YEADSP ; 年データ表示
[HL]=#DIV_SIG (A) ; /
HL++
X=CE_MOND (A) ; 月データの取り込み
CALL !MONDSP ; 月データ表示
[HL]=#DIV_SIG (A) ; /
HL++
X=CE_DAYD (A) ; 日データの取り込み
CALL !MONDSP ; 日データ表示
HL=#LCDDS+30 ; カーソル・アドレスの設定
DE=#OPEN_D
CALL !DTILSET ; 0 0 0 0 H + 3 0 ( 2 行目) に設定
SET1 DS_BRAF
SET1 LCDCHGF
SET1 CR_DSPF
CUR_ADR=#41
INV_DS=#0 ; 反転表示削除
SET1 IV_CHGF
endif
CALL !SC_RKY0
break
case 1:
if_bit(MD_CG4F)

```

```

        CLR1 MD_CG4F
        HL=#LCDDS+60
        DE=#END_D
        CALL !DTILSET
        SET1 LCDCHGF
        SET1 DS_ERAF
        CUR_ADR=#71
    endif
    CALL !SC_RKY1
    break
case 2:
    if_bit(MD_CG4F)
        CLR1 MD_CG4F
        HL=#LCDDS+90
        DE=#ALARM_D
        CALL !DTILSET
        !LCDDS+101=#PAR_LE (A)
        !LCDDS+104=#PAR_RI (A)
        SET1 DS_ERAF
        SET1 LCDCHGF
        CLR1 CR_DSPF
    endif
    CALL !SC_RKY2
    break
case 3:
    if_bit(MD_CG4F)
        CLR1 MD_CG4F
        HL=#LCDDS+120
        DE=#BUSI_D
        CALL !DTILSET
        SET1 DS_ERAF
        SET1 LCDCHGF
        SET1 CR_DSPF
        CUR_ADR=#131
    endif
    CALL !SC_RKY3
    break
case 4:
    if_bit(MD_CG4F)
        CLR1 MD_CG4F
        HL=#LCDDS+150
        DE=#MEMO_D1
        CALL !DTILSET
        SET1 LCDCHGF
        SET1 DS_ERAF
        CUR_ADR=#161
    endif
    CALL !SC_RKY4
    break
case 5:
    if_bit(MD_CG4F)
        CLR1 MD_CG4F
        CLR1 CR_DSPF
        CALL !LCD_DSC
        HL=#LCDDS+90
        DE=#MEM_OV
        CALL !DTILSET
        SET1 LCDCHGF
    endif
    CALL !SC_RKY5
    break
ends
RET

;*****
; スケジュール・データ登録キー入力処理
;*****
SC_RKY0:
    IF_BIT(KY_SIGF)
        CLR1 KY_SIGF
        if_bit(DS_ERAF)
            CLR1 DS_ERAF
            !LCDDS+96=#SPC (A)
        endif
        IF(KEY_CD<#'0' || KEY_CD>#'9')

```

```

switch(KEY_CD)
case DELETE:
    HL=#LCDDS
    B=CUR_ADR (A)
    [HL+B]=#SPC (A)           ;カーソル・アドレスにスペースを設定
    break
case CLEAR:                   ;前モードに設定
    MODE3=#0
    SET1 MD_CG3F
    FOR_ADR=#ADRSTA
    FOR_BNK=#0
    MODE4=#0
    SET1 MD_CG4F
    break
case ENTER:
    CALL !OPN_TIM
    if_bit(TIM_OKF)
    MODE4=#1                 ;終了時間入力モードに設定
    SET1 MD_CG4F
    endif
    break
case CUR_RI:
    if(CUR_ADR==#42)         ;時データ下位アドレス
        CUR_ADR=#44         ;分データ上位アドレスに設定
    elseif(CUR_ADR!="#45)   ;分データ下位アドレスでない
        CUR_ADR++          ;アドレス値をインクリメント
    endif
    break
case CUR_LE:
    if(CUR_ADR==#44)         ;分データ上位アドレス
        CUR_ADR=#42         ;時データ下位アドレスに設定
    elseif(CUR_ADR!="#41)   ;時データ上位アドレスでない
        CUR_ADR--          ;アドレス値をデクリメント
    endif
endif
ends
else
    B=CUR_ADR (A)           ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)       ;数値データを設定
    if(CUR_ADR==#42)         ;時データ下位アドレス
        CUR_ADR=#44         ;分データ上位アドレスに設定
    elseif(CUR_ADR!="#45)   ;分データ下位アドレスでない
        CUR_ADR++          ;アドレス値をインクリメント
    endif
endif
ENDIF
ENDIF
RET

SC_RKY1:
IF_BIT(KY_SIGF)             ;キー入力あり
CLR1 KY_SIGF
if_bit(DS_ERAF)             ;1ライン・データ削除
CLR1 DS_ERAF
!LCDDS+65=#SPC (A)         ;カーソル・アドレスの設定
endif
IF(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
switch(KEY_CD)
case DELETE:
    HL=#LCDDS
    B=CUR_ADR (A)
    [HL+B]=#SPC (A)           ;カーソル・アドレスにスペースを設定
    break
case CLEAR:                   ;前モードに設定
    MODE3=#0
    SET1 MD_CG3F
    FOR_ADR=#ADRSTA
    FOR_BNK=#0
    MODE4=#0
    SET1 MD_CG4F
    break
case ENTER:
    CALL !END_TIM
    if_bit(TIM_OKF)
    MODE4=#2                 ;アラーム設定モードに設定
    SET1 MD_CG4F

```

```

        endif
        break
    case CUR_RI:
        if(CUR_ADR==#72)           ;時データ下位アドレス
            CUR_ADR=#74           ;分データ上位アドレスに設定
        elseif(CUR_ADR1=#75)      ;分データ下位アドレスでない
            CUR_ADR++             ;アドレス値をインクリメント
        endif
        break
    case CUR_LE:
        if(CUR_ADR==#74)           ;分データ上位アドレス
            CUR_ADR=#72           ;時データ下位アドレスに設定
        elseif(CUR_ADR1=#71)      ;時データ上位アドレスでない
            CUR_ADR--             ;アドレス値をデクリメント
        endif
    ends
else
    B=CUR_ADR (A)                 ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)             ;数値データを設定
    if(CUR_ADR==#72)             ;時データ下位アドレス
        CUR_ADR=#74             ;分データ上位アドレスに設定
    elseif(CUR_ADR1=#75)         ;分データ下位アドレスでない
        CUR_ADR++               ;アドレス値をインクリメント
    endif
ENDIF
ENDIF
RET

SC_RKY2:
if_bit(KY_SIGF)                 ;キー入力あり
    CLR1 KY_SIGF
    if_bit(DS_ERAF)              ;1ライン・データ削除
        CLR1 DS_ERAF
        !LCDDS+97=#SPC (A)       ;カーソル・アドレスの設定
        CUR_ADR=#101
    endif
    switch(KEY_CD)
    case CLEAR:                  ;前モードに設定
        MODE3=#0
        SET1 MD_CG3F
        FOR_ADR=#ADRSTA
        FOR_BNK=#0
        MODE4=#0
        SET1 MD_CG4F
        break
    case ENTER:                  ;用件入力モードに設定
        MODE4=#3
        SET1 MD_CG4F
        break
    case CUR_LE:
        !LCDDS+101=#PAR_LE (A)   ;ONの(を設定
        !LCDDS+104=#PAR_RI (A)   ;ONの)を設定
        !LCDDS+106=#SPC (A)       ;OFFを非選択に設定
        !LCDDS+110=#SPC (A)
        break
    case CUR_RI:
        !LCDDS+101=#SPC (A)       ;ONを非選択に設定
        !LCDDS+104=#SPC (A)
        !LCDDS+106=#PAR_LE (A)   ;OFFの(を設定
        !LCDDS+110=#PAR_RI (A)   ;OFFの)を設定
    ends
endif
RET

SC_RKY3:
IF_BIT(KY_SIGF)                 ;キー入力あり
    CLR1 KY_SIGF
    if_bit(DS_ERAF)              ;1ライン・データ削除
        CLR1 DS_ERAF
        !LCDDS+130=#SPC (A)       ;カーソル・アドレスの設定
    endif
    if(KEY_CD<#SPC || KEY_CD>#`2`) ;文字コード以外?
        switch(KEY_CD)
        case DELETE:

```

```

B=CUR_ADR (A) ;現在のカーソル・アドレスを設定
C=CUR_ADR (A) ;データ転送対象のアドレス(+1)を設定
C++
HL=#LCDDS
while(C<#150) (A)
  [HL+B]=[HL+C] (A) ;データ転送
  B++
  C++
endw
[HL+B]=#SPC (A) ;行最終アドレスにスペース・データを設定
break
case CLEAR: ;前モードに設定
  MODE3=#0
  SET1 MD_CG3F
  FOR_ADR=#ADRSTA
  FOR_BNK=#0
  MODE4=#0
  SET1 MD_CG4F
  break
case ENTER: ;メモ入力モードに設定
  MODE4=#4
  SET1 MD_CG4F
  break
case CUR_RI:
  if(CUR_ADR<#149) ;カーソル・アドレス5行目最終行以内
    CUR_ADR++
  endif
  break
case CUR_LE:
  if(CUR_ADR>#131) ;カーソル・アドレス5行目先頭アドレス以上
    CUR_ADR--
  endif
ends
else
  B=CUR_ADR (A) ;カーソル・アドレスの設定
  HL=#LCDDS
  [HL+B]=KEY_CD (A) ;数値データを設定
  if(CUR_ADR<#149) ;カーソル・アドレス5行目最終行以内
    CUR_ADR++
  endif
endif
ENDIF
RET
SC_RKY4:
IF_BIT(KY_SIGF) ;キー入力あり
CLR1 KY_SIGF
if_bit(DS_ERAF) ;1ライン・データ削除
CLR1 DS_ERAF
|LCDDS+156=#SPC (A) ;カーソル・アドレスの設定
endif
IF(KEY_CD<#SPC || KEY_CD>#z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
  B=CUR_ADR (A) ;現在のカーソル・アドレスを設定
  C=CUR_ADR (A) ;データ転送対象のアドレス(+1)を設定
  C++
  HL=#LCDDS
  while(C<#240) (A) ;データ転送
    [HL+B]=[HL+C] (A)
    B++
    C++
  endw
  [HL+B]=#SPC (A) ;行最終アドレスにスペース・データを設定
  break
case S_DEL: ;1ライン・データ削除
  if(CUR_ADR<#180)
    HL=#LCDDS+150 ;カーソル・アドレスの設定
    DE=#MEMO_DI
    CALL 1DT1LSET
    CUR_ADR=#161 ;カーソル・アドレスを先頭に設定
  elseif(CUR_ADR<#210)
    HL=#LCDDS+180 ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL 1DT1LSET

```

```

CUR_ADR=#180          ;カーソル・アドレスを先頭に設定
else
HL=#LCDDS+210        ;カーソル・アドレスの設定
DE=#CLR1LD
CALL !DT1LSET
CUR_ADR=#210          ;カーソル・アドレスを先頭に設定
endif
break
case CLEAR:           ;前モードに設定
MODE3=#0
SET1 MD.CG3F
FOR_ADR=#ADRSTA
FOR_BNK=#0
MODE4=#0
SET1 MD.CG4F
break
CASE ENTER:          ;メモ入力モードに設定
M_BANK=#3
CALL !BANKSET
A=!0BF00H
IF(A=#VAC_DAT)      ;最終アドレスのメモリ空きあり
CLR1 ADSEAF
M_BANK=#0
CALL !BANKSET
H=#ADRSTA
L=#0
B=#SCA_DAT          ;比較データをスケジュール・データに設定
C=#16                ;1ライン比較の開始アドレスを設定(HL+C)
DE=#LCDDS            ;比較先頭アドレス
CALL !SAVADR        ;登録メモリ・アドレスを設定
DE=#DTWORK          ;転送用データの作成
[DE]=#SCA_DAT(A)    ;データ種類の設定
DE++
[DE]=CE_YEAD(A)     ;スケジュール年データ設定
DE++
[DE]=CE_MOND(A)    ;スケジュール月データ設定
DE++
[DE]=CE_DAYD(A)    ;スケジュール日データ設定
DE++
[DE]=CE_TIME+1(A)  ;スケジュール開始時間時データ設定
DE++
[DE]=CE_TIME(A)    ;スケジュール開始時間分データ設定
DE++
[DE]=CE_TIME+3(A)  ;スケジュール終了時間時データ設定
DE++
[DE]=CE_TIME+2(A)  ;スケジュール終了時間分データ設定
DE++
A=!LCDDS+101        ;ON表示(アドレスデータ参照)
if(A=#PAR_LE)       ; (あり)
[DE]=#ALWON(A)      ;アラームONデータ設定
else
[DE]=#ALNOFF(A)     ;アラームOFFデータ設定
endif
DE++
A=#0
for(i=#9;i<#10H;i++) ;9-15バイトのデータをクリア
[DE]=A
DE++
next
HL=#LCDDS           ;LCD表示データをワーク・エリアに設定
for(i=#0;i<#240;i++)
[DE]=[HL](A)
DE++
HL++
next
DE=#DTWORK
H=#MENPAG(A)
L=#0
CALL !DT_SAV        ;データ登録
MODE3=#0            ;登録終了
SET1 MD.CG3F
FOR_ADR=#ADRSTA
FOR_BNK=#0
MODE4=#0
SET1 MD.CG4F

```



```

ELSE
    SET1    MD_CG4F    ;メモリ・アドレスに空きなし
    MODE4=#5          ;オーバフロー・モードに設定
ENDIF
break
case CUR_RI:
    if(CUR_ADR<#239)  ;カーソル・アドレス5行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#161)  ;カーソル・アドレス5行目先頭アドレス以上
        CUR_ADR--
    endif
    break
case CUR_UP:
    if(CUR_ADR>=#191) ;カーソル・アドレスを1行上(+30)に設定
        CUR_ADR+=#30
    endif
    break
case CUR_DN:
    if(CUR_ADR<#210) ;カーソル・アドレスを1行下(-30)に設定
        CUR_ADR-=#30
    endif
ends
ELSE
    B=CUR_ADR (A)      ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A) ;数値データを設定
    if(CUR_ADR<#239)  ;カーソル・アドレス5行目最終行以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

SC_RKY5:
    if_bit(KY_SIGF)
    CLR1  KY_SIGF
    switch(KEY_CD)
    case CLEAR:          ;前モードに設定
        MODE3=#0
        SET1  MD_CG3F
    ends
endif
RET

OPN_TIM:
    CLR1  TIM_OKF
    X=!LCDDS+45 (A)      ;開始時間分データの取り込み
    A=!LCDDS+44
    CALL  !ASC_DEL
    if(A<#60H)          ;分データ60分未満
        CE_TIME=A
        X=!LCDDS+42 (A) ;開始時間時データの取り込み
        A=!LCDDS+41
        CALL  !ASC_DEL
        if(A<#24H)      ;時データ24未満
            CE_TIME+1=A
            SET1  TIM_OKF ;入力データ正常値
        endif
    endif
RET

END_TIM:
    CLR1  TIM_OKF
    X=!LCDDS+75 (A)     ;終了時間分データの取り込み
    A=!LCDDS+74
    CALL  !ASC_DEL
    if(A<#60H)          ;分データ60分未満
        CE_TIME+2=A
        X=!LCDDS+72 (A) ;終了時間時データの取り込み
        A=!LCDDS+71
        CALL  !ASC_DEL

```

```

        if(A<#24H)                                ;時データ24未満
            CE_TIME+3=A
            SET1 TIM_OKF                          ;入力データ正常値
        endif
    endif
    RET

;*****
;   スケジュール修正モード
;*****
SC_COR:
    if_bit(MD_CG4F)                              ;修正データの変更
        CLR1 MD_CG4F
        M_BANK=MEMBNK (A)                       ;メモリ・アドレスの設定
        CALL !BANKSET
        D=MEMPAG (A)
        E=#16
        HL=#LCDDS                               ;LCD表示アドレスの設定
        CALL !DTIPSET                          ;表示データの転送
        CUR_ADR=#41                             ;カーソル・アドレスを1文字目に設定
        SET1 CR_DSPF
        INV_DS=#0
        SET1 IV_CHGF
        SET1 LCDCHGF
    endif
    switch(MODE4)
    case 0:
        CALL !SC_CKY0
        break
    case 1:
        CALL !SC_CKY1
        break
    case 2:
        CALL !SC_CKY2
        break
    case 3:
        CALL !SC_CKY3
        break
    case 4:
        CALL !SC_CKY4
    ends
    RET

;*****
;   スケジュール・データ修正キー入力処理
;*****
SC_CKY0:
    IF_BIT(KY_SIGF)                              ;キー入力あり
        CLR1 KY_SIGF
        IF(KEY_CD<#'0' || KEY_CD>#'9')          ;数字コード以外?
            switch(KEY_CD)
            case DELETE:
                HL=#LCDDS
                B=CUR_ADR (A)
                [HL+B]=#SPC (A)                 ;カーソル・アドレスにスペースを設定
                break
            case CLEAR:
                MODE3=#0
                SET1 MD_CG3F
                FOR_ADR=#ADRSTA
                FOR_BNK=#0
                MODE4=#0
                SET1 MD_CG4F
                break
            case ENTER:
                CALL !OPN_TIM
                if_bit(TIM_OKF)
                CALL !CAL_CSV
                endif
                break
            case CUR_RI:
                if(CUR_ADR==#42)                 ;時データ下位アドレス
                    CUR_ADR=#44                ;分データ上位アドレスに設定
                elseif(CUR_ADR!="#45)           ;分データ下位アドレスでない
                    CUR_ADR++                  ;アドレス値をインクリメント

```

```

        endif
        break
    case CUR_LE:
        if(CUR_ADR==#44)           ;分データ上位アドレス
            CUR_ADR=#42           ;時データ下位アドレスに設定
        elseif(CUR_ADR!="#41)     ;時データ上位アドレスでない
            CUR_ADR--            ;アドレス値をデクリメント
        endif
        break
    case CUR_DN:
        CALL !OPN_TIM
        if_bit(TIM_OKF)
        MODE4=#1
        CUR_ADR+=#30
    endif
ends
ELSE
B=CUR_ADR (A)                   ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)               ;数値データを設定
if(CUR_ADR==#42)                ;時データ下位アドレス
    CUR_ADR=#44                 ;分データ上位アドレスに設定
elseif(CUR_ADR!="#45)          ;分データ下位アドレスでない
    CUR_ADR++                   ;アドレス値をインクリメント
endif
ENDIF
ENDIF
RET

SC_CKY1:
IF_BIT(XY_SIGF)                 ;キー入力あり
CLR1 KY_SIGF
IF(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
    switch(KEY_CD)
    case DELETE:
        HL=#LCDDS
        B=CUR_ADR (A)
        [HL+B]=#SPC (A)         ;カーソル・アドレスにスペースを設定
        break
    case CLEAR:
        MODE3=#0
        SET1 MD_CG3F
        FOR_ADR=#ADRSTA
        FOR_BNK=#0
        MODE4=#0
        SET1 MD_CG4F
        break
    case ENTER:
        CALL !END_TIM
        if_bit(TIM_OKF)
        CALL !CAL_CSV
        endif
        break
    case CUR_RI:
        if(CUR_ADR==#72)         ;時データ下位アドレス
            CUR_ADR=#74         ;分データ上位アドレスに設定
        elseif(CUR_ADR!="#75)   ;分データ下位アドレスでない
            CUR_ADR++          ;アドレス値をインクリメント
        endif
        break
    case CUR_LE:
        if(CUR_ADR==#74)         ;分データ上位アドレス
            CUR_ADR=#72         ;時データ下位アドレスに設定
        elseif(CUR_ADR!="#71)   ;時データ上位アドレスでない
            CUR_ADR--          ;アドレス値をデクリメント
        endif
        break
    case CUR_UP:
        CALL !END_TIM
        if_bit(TIM_OKF)
        MODE4=#0
        CUR_ADR--=#30
        endif
        break
    case CUR_DN:

```

```

        CALL !END_TIM
        if_bit(TIM_OKF)
            MODE4=#2 ;アラーム時間修正モードに設定
            CLRI CR_DSPF
        endif
    ends
ELSE
    B=CUR_ADR (A) ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A) ;数値データを設定
    if(CUR_ADR==#72) ;時データ下位アドレス
        CUR_ADR=#74 ;分データ上位アドレスに設定
    elseif(CUR_ADR!=#75) ;分データ下位アドレスでない
        CUR_ADR++; ;アドレス値をインクリメント
    endif
ENDIF
ENDIF
RET

SC_CK2:
if_bit(KY_SIGF) ;キー入力あり
    CLRI KY_SIGF
    switch(KEY_CD)
        case CLEAR: ;前モードに設定
            MODE3=#0
            SETI MD_CG3F
            FOR_ADR=#ADRSTA
            FOR_BNK=#0
            MODE4=#0
            SETI MD_CG4F
            break
        case ENTER: ;修正データを登録
            CALL !CAL_CSV
            break
        case CUR_LE:
            !LCDDS+101=#PAR_LE (A) ;ONの(を設定
            !LCDDS+104=#PAR_RI (A) ;ONの)を設定
            !LCDDS+106=#SPC (A) ;OFFを非選択に設定
            !LCDDS+110=#SPC (A)
            break
        case CUR_RI:
            !LCDDS+101=#SPC (A) ;ONを非選択に設定
            !LCDDS+104=#SPC (A)
            !LCDDS+106=#PAR_LE (A) ;OFFの(を設定
            !LCDDS+110=#PAR_RI (A) ;OFFの)を設定
            break
        case CUR_UP:
            MODE4=#1 ;終了時間修正モードに設定
            CUR_ADR=#71
            SETI CR_DSPF
            break
        case CUR_DN:
            MODE4=#3 ;用件修正モードに設定
            CUR_ADR=#131
            SETI CR_DSPF
    ends
endif
RET

SC_CK3:
IF_BIT(KY_SIGF) ;キー入力あり
    CLRI KY_SIGF
    IF(KEY_CD<#SPC || KEY_CD>#z') ;文字コード以外?
        switch(KEY_CD)
            case DELETE:
                B=CUR_ADR (A) ;現在のカーソル・アドレスを設定
                C=CUR_ADR (A) ;データ転送対象のアドレス(+1)を設定
                C++
                HL=#LCDDS
                while(C<#150) (A)
                    [HL+B]=[HL+C] (A) ;データ転送
                    B++
                    C++
                endwhile
                [HL+B]=#SPC (A) ;行最終アドレスにスペース・データを設定
            endswitch
        endswitch
    endif
endif

```

```

        break
case CLEAR:                                ;前モードに設定
        MODE3=#0
        SET1 MD_CG3F
        FOR_ADR=#ADRSTA
        FOR_BNK=#0
        MODE4=#0
        SET1 MD_CG4F
        break
case ENTER:                                ;修正データを登録
        CALL !CAL_CSV
        break
case CUR_RI:                                ;カーソル・アドレス5行目最終行以内
        if(CUR_ADR<#149)
                CUR_ADR++
        endif
        break
case CUR_LE:                                ;カーソル・アドレス5行目先頭アドレス以上
        if(CUR_ADR>#131)
                CUR_ADR--
        endif
        break
case CUR_UP:                                ;アラーム時間修正モードに設定
        MODE4=#2
        CLR1 CR_DSPF
        break
case CUR_DN:                                ;メモ修正モードに設定
        MODE4=#4
        CUR_ADR+=#30
ends
ELSE
        B=CUR_ADR (A)                        ;カーソル・アドレスの設定
        HL=#LCDDS
        [HL+B]=KEY_CD (A)                    ;数値データを設定
        if(CUR_ADR<#149)                    ;カーソル・アドレス5行目最終行以内
                CUR_ADR++
        endif
ENDIF
ENDIF
RET

SC_CKY4:
        IF_BIT(KY_SIGF)                      ;キー入力あり
        CLR1 KY_SIGF
        IF(KEY_CD<#SPC || KEY_CD>#z')        ;文字コード以外?
                switch(KEY_CD)
                case DELETE:
                        B=CUR_ADR (A)        ;現在のカーソル・アドレスを設定
                        C=CUR_ADR (A)        ;データ転送対象のアドレス(+1)を設定
                        C++
                        HL=#LCDDS
                        while(C<#240) (A)
                                [HL+B]=[HL+C] (A) ;データ転送
                                B++
                                C++
                        endw
                        [HL+B]=#SPC (A)      ;行最終アドレスにスペース・データを設定
                        break
                case S_DEL:                   ;1ライン・データ削除
                        if(CUR_ADR<#179)
                                HL=#LCDDS+150 ;カーソル・アドレスの設定
                                DE=#MEMO_D1
                                CALL !DT1LSET
                                CUR_ADR=#161 ;カーソル・アドレスを先頭に設定
                        elseif(CUR_ADR<#209)
                                HL=#LCDDS+180 ;カーソル・アドレスの設定
                                DE=#CLR1LD
                                CALL !DT1LSET
                                CUR_ADR=#180 ;カーソル・アドレスを先頭に設定
                        else
                                HL=#LCDDS+210 ;カーソル・アドレスの設定
                                DE=#CLR1LD
                                CALL !DT1LSET
                                CUR_ADR=#210 ;カーソル・アドレスを先頭に設定
                        endif
endif

```

```

        break
case CLEAR:                                ;前モードに設定
    MODE3=#0
    SET1 MD_CG3F
    FOR_ADR=#ADRSTA
    FOR_BNK=#0
    MODE4=#0
    SET1 MD_CG4F
    break
case ENTER:                                ;修正データを登録
    CALL !CAL_CSV
    break
case CUR_RI:                                ;カーソル・アドレス5行目最終行以内
    if(CUR_ADR<#239)
        CUR_ADR++
    endif
    break
case CUR_LE:                                ;カーソル・アドレス5行目先頭アドレス以上
    if(CUR_ADR<#161)
        CUR_ADR--
    endif
    break
case CUR_UP:                                ;カーソル・アドレスを1行上(+30)に設定
    if(CUR_ADR<#191)
        CUR_ADR-=#30
    elseif(CUR_ADR<#180)
        CUR_ADR-=#30
        MODE4=#3                                ;用件修正モードに設定
    endif
    break
case CUR_DN:                                ;カーソル・アドレスを1行下(-30)に設定
    if(CUR_ADR<#210)
        CUR_ADR+=#30
    endif
ends
ELSE
    B=CUR_ADR (A)                                ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)                            ;数値データを設定
    if(CUR_ADR<#239)
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

;*****
; カレンダ指定月表示
;*****
CLE_DSP:
    if_bit(MD_CG2F)
        CLR1 MD_CG2F
        INV_DS=#0                                ;反転表示行削除
        SET1 IV_CHGF
        HL=#LCDDS                                ;カレンダ・モード選択表示の設定
        DE=#CLR1LD                                ;LCD1行目のデータを設定
        CALL !DTILSET

        DE=#YEAR_D1                                ;LCD2行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD                                ;LCD3行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD                                ;LCD4行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD                                ;LCD5行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD                                ;LCD6行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD                                ;LCD7行目のデータを設定
        CALL !DTILSET

```

```

DE=#CLRILD ;LCD 8行目のデータを設定
CALL !DTILSET
CUR_ADR=#60
SET1 CR_DSPF
SET1 LCDCGPF
endif
switch(MODE2)
case 0:
CALL !CE_DSET ;表示月設定モード
break
case 1:
CALL !CE_DCHG ;表示月変更モード
ends
if_bit(CLDCGPF) ;カレンダー表示月変更あり
CLR1 CLDCGPF
CLR1 CR_DSPF
CALL !LCD_DSC
CALL !CLDDSP ;カレンダー月表示
CALL !SCAMAK ;スケジュール・マーク表示
CALL !W_HDSP ;週休表示
CALL !HOLDSP ;指定休日表示
endif
RET

;*****
; 表示月設定処理
;*****
CE_DSET:
switch(MODE3)
CASE 0:
if_bit(KY_SIGF) ;キー入力あり
CLR1 KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
switch(KEY_CD)
case DELETE:
HL=#LCDDS
B=CUR_ADR (A)
[HL+B]=#SPC (A) ;カーソル・アドレスにスペースを設定
break
case CLEAR: ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER: ;年データの決定
HL=#LCDDS+60
CALL !YDSCNV
MODE3=#1
SET1 MD_CG3F
break
case CUR_RI:
if(CUR_ADR<#63) ;上位データ設定
CUR_ADR++ ;1桁下位にカーソルを設定
endif
break
case CUR_LE:
if(CUR_ADR>#60) ;下位データ設定
CUR_ADR-- ;1桁上位にカーソルを設定
endif
ends
else
B=CUR_ADR (A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A) ;数値データを設定
if(CUR_ADR<#63) ;設定データ最下位桁以内
CUR_ADR++
endif
endif
endif
break
CASE 1:
if_bit(MD_CG3F)
CLR1 MD_CG3F
HL=#LCDDS+90
DE=#MON_D1 ;LCD 8行目のデータを設定

```

```

CALL !DTILSET
CUR_ADR=#120
SET1 LCDCHGF
endif
IF_BIT(KY_SIGF) ;キー入力あり
CLR1 KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
switch(KEY_CD)
case DELETE:
HL=#LCDDS
B=CUR_ADR (A)
[HL+B]=#SPC (A) ;カーソル・アドレスにスペースを設定
break
case CLEAR: ;前モードに設定
MODE1=#0
SET1 MD_CG1F
break
case ENTER: ;月データの決定
HL=#LCDDS+120
CALL !MDSCHV
MODE2=#1
SET1 CLDCGF
break
case CUR_R1:
if(CUR_ADR<#121) ;上位データ設定
CUR_ADR++ ;1桁下位にカーソルを設定
endif
break
case CUR_LE:
if(CUR_ADR>#120) ;下位データ設定
CUR_ADR-- ;1桁上位にカーソルを設定
endif
endif
ends
else
B=CUR_ADR (A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A) ;数値データを設定
if(CUR_ADR<#121) ;設定データ最下位桁以内
CUR_ADR++
endif
endif
endif
ENDIF
ends
RET

;*****
; 表示月変更処理
;*****
CE_DCHG:
if_bit(KY_SIGF) ;キー入力あり
CLR1 KY_SIGF
switch(KEY_CD)
case UP: ;表示月に+1カ月
CALL !NEX_MON
break
case DOWN: ;表示月から-1カ月
CALL !BEF_MON
break
case S_UP: ;表示月に+1年
CALL !NEX_YEA
break
case S_DOWN: ;表示月から-1年
CALL !BEF_YEA
break
case CLEAR: ;カレンダー・モード選択に設定
MODE1=#0
SET1 MD_CG1F
ends
endif
RET

;*****
; 休日設定処理
;*****
CLE_HOL:

```



```

switch(MODE2)
case 0:                                     ;休日設定のモード選択
    CALL    ICE_HSEL
    break
case 1:                                     ;週休設定
    CALL    ICE_HWEX
    break
case 2:                                     ;日付指定の休日設定
    CALL    ICE_HDAY
ends
RET

;*****
;    休日設定のモード選択
;*****
CE_HSEL:
    if_bit(MD_CG2F)
        CLR1    MD_CG2F
        HL=#LCDDS                                     ;カレンダー・モード選択表示の設定
        DE=#SETHOL2                                 ;LCD 1行目のデータを設定
        CALL    !DTILSET

        DE=#WEKHOL1                                 ;LCD 2行目のデータを設定
        CALL    !DTILSET

        DE=#SETDAT1                                 ;LCD 3行目のデータを設定
        CALL    !DTILSET

        DE=#CLR1LD                                  ;LCD 4行目のデータを設定
        CALL    !DTILSET

        DE=#CLR1LD                                  ;LCD 5行目のデータを設定
        CALL    !DTILSET

        DE=#CLR1LD                                  ;LCD 6行目のデータを設定
        CALL    !DTILSET

        DE=#CLR1LD                                  ;LCD 7行目のデータを設定
        CALL    !DTILSET

        DE=#CLR1LD                                  ;LCD 8行目のデータを設定
        CALL    !DTILSET

        SET1    IV_CHGF                               ;反転表示行を2行目に設定
        INV_DS=#2
        SET1    LCDCHGF
    endif

    if_bit(KY_SIGF)
        CLR1    KY_SIGF
        switch(KEY_CD)                               ;キー・コード分岐
        case CUR_UP:
            if(INV_DS>#2)
                INV_DS--
                SET1    IV_CHGF                     ;反転表示行変更状態に設定
            endif
            break
        case CUR_DN:
            if(INV_DS<#3)
                INV_DS++
                SET1    IV_CHGF                     ;反転表示行変更状態に設定
            endif
            break
        case CLEAR:
            MODE1=#0
            SET1    MD_CG1F                           ;前面モードに変更
            break
        case ENTER:
            A=INV_DS
            A--
            MODE2=A
            MODE3=#0
            SET1    MD_CG2F                           ;モード変更状態に設定
            SET1    MD_CG3F
        ends

```

```

endif
RET

;*****
;  週休設定処理
;*****
CE_HWEK:
  if_bit(MD_CG2F)
    CLR1  MD_CG2F
    HL=#LCDDS                ;カレンダー・モード選択表示の設定
    DE=#WEKHOL2            ;LCD1行目のデータを設定
    CALL  !DTILSET

    DE=#WEEK_D              ;LCD2行目のデータを設定
    CALL  !DTILSET

    DE=#CLR1LD              ;LCD3行目のデータを設定
    CALL  !DTILSET

    DE=#CLR1LD              ;LCD4行目のデータを設定
    CALL  !DTILSET

    DE=#HOLNAK              ;LCD5行目のデータを設定
    CALL  !DTILSET

    DE=#WEEKNAK             ;LCD6行目のデータを設定
    CALL  !DTILSET

    DE=#CLR1LD              ;LCD7行目のデータを設定
    CALL  !DTILSET

    DE=#CLR1LD              ;LCD8行目のデータを設定
    CALL  !DTILSET

    SET1  IV_CHGF            ;反転表示行をクリア
    INV_DS=#0
    CUR_ADR=#62              ;カーソル・アドレスの設定
    SET1  CR_DSPF
    SET1  LCDCHGF

    HL=#LCDDS+62            ;休日表示先頭アドレス
    X=#WEKHOLD (A)          ;週休データの取り込み
    for(i=#0;i<#7;i++)
      A=X
      A>>=1                ;曜日データの取り込み
      X=A
      if_bit(CY)            ;指定曜日休日
        A=#'o'              ;休日表示
      else
        A=#'x'              ;平日表示
      endif
      [HL]=A                ;表示データの設定
      HL++                  ;次の曜日に設定(+4)
      HL++
      HL++
      HL++
    next
  endif

  IF_BIT(KY_SIGF)
    CLR1  KY_SIGF
    switch(KEY_CD)          ;キー・コード分岐
    case CUR_UP:
      HL=#LCDDS            ;現在のアドレス取り込み
      B=CUR_ADR (A)
      [HL+B]=#'o' (A)      ;休日データの設定
      break
    case CUR_DN:
      HL=#LCDDS            ;現在のアドレス取り込み
      B=CUR_ADR (A)
      [HL+B]=#'x' (A)      ;平日データの設定
      break
    case CUR_RI:
      if(CUR_ADR)>=#86)    ;カーソル位置=S A T
        CUR_ADR=#62      ;カーソル位置=S U Nに設定
      endif
  endif

```

```

else
    CUR_ADR+=#4          ;カーソルを次の曜日に設定
endif
break
case CUR_LE:
    if(CUR_ADR<=#62)    ;カーソル位置 = S U N
        CUR_ADR=#86    ;カーソル位置 = S A T に設定
    else
        CUR_ADR-=#4    ;カーソルを前の曜日に設定
    endif
    break
case CLEAR:
    MODE1=#0           ;前画面モードに変更
    SET1 MD_CG1F
    break
case ENTER:
    HL=#LCDDS+62       ;休日データの先頭アドレスを設定
    X=#0
    for(i=#0;i<#7;i++)
        if([HL]=#'o') (A) ;休日データ
            SET1 CY      ;C Y フラグをセット
        else
            CLR1 CY      ;C Y フラグをクリア
        endif
        A=X
        RORC A,1         ;曜日データの取り込み
        X=A
        HL++            ;次のデータにアドレスを設定
        HL++
        HL++
        HL++
    next
    RORC A,1
    WEKHOLD=A
    MODE1=#0
    SET1 MD_CG1F
ends
ENDIF
RET

;*****
; 指定休日設定処理
;*****
CE_HDAY:
switch(MODE3)
case 0: ;休日年設定モード
    CALL !HOL_Y
    break
case 1: ;休日月設定モード
    CALL !HOL_M
    break
case 2: ;休日日設定モード
    CALL !HOL_D
    break
case 3:
    if_bit(MD_CG3F)
        CLR1 MD_CG3F
        CALL !LCD_DSC
        HL=#LCDDS+90 ;カーソル・アドレスの設定
        DE=#MEM_OV   ;エラー表示
        CALL !DT1LSET ;0000H+90(4行目)に設定
    endif
    CALL !HOL_OV
ends
RET

;*****
; 休日年設定処理
;*****
HOL_Y:
    if_bit(MD_CG3F)
        CLR1 MD_CG3F
        CALL !LCD_DSC ;LCD表示クリア
        SET1 !V_CHGF  ;反転表示行をクリア
        INV_DS=#0

```

```

HL=#LCDDS
DE=#SBTDAT2           ;LCD 1行目のデータを設定
CALL  !DTILSET
DE=#YEAR_D2           ;LCD 2行目のデータを設定
CALL  !DTILSET
CUR_ADR=#39
SETI  CR_DSPF
SETI  LCDCHGF
endif

IF_BIT(KY_SIGF)
CLR1  KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
switch(KEY_CD) ;キーコード分岐
case CLEAR:
MODE1=#0 ;前画面モードに変更
SETI  MD_CG1F
break
case DELETE:
HL=#LCDDS
B=CUR_ADR (A)
[HL+B]=#SPC (A) ;カーソル・アドレスにスペースを設定
break
case CUR_RI:
if(CUR_ADR<#42) ;年データ4桁目入力
CUR_ADR++ ;カーソル・アドレスに1加算
endif
break
case CUR_LE:
if(CUR_ADR>#39) ;年データ1桁目入力
CUR_ADR-- ;カーソル・アドレスから1減算
endif
break
case ENTER: ;表示年データの登録
HL=#LCDDS+39
CALL  !YDSCHV ;表示データを数値に変換
MODE3=#1
SETI  MD_CG3F ;モード変更状態に設定
ends
else
B=CUR_ADR (A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A) ;数値データを設定
if(CUR_ADR<#42) ;設定データ最下位桁以内
CUR_ADR++
endif
endif
endif
endif
RET

;*****
; 休日設定処理
;*****
HOL_M:
if_bit(MD_CG3F)
CLR1  MD_CG3F

HL=#LCDDS+60
DE=#MON_D2           ;LCD 3行目のデータを設定
CALL  !DTILSET
CUR_ADR=#69
SETI  LCDCHGF
endif

IF_BIT(KY_SIGF)
CLR1  KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
switch(KEY_CD) ;キーコード分岐
case CLEAR:
MODE1=#0 ;前画面モードに変更
SETI  MD_CG1F
break
case DELETE:
HL=#LCDDS

```

```

        B=CUR_ADR (A)
        [HL+B]=#SPC (A)          ;カーソル・アドレスにスペースを設定
        break
    case CUR_RI:
        if(CUR_ADR<#70)         ;月データ2桁目入力
            CUR_ADR++          ;カーソル・アドレスに1加算
        endif
        break
    case CUR_LE:
        if(CUR_ADR>#69)         ;月データ1桁目入力
            CUR_ADR--          ;カーソル・アドレスから1減算
        endif
        break
    case ENTER:
        HL=#LCDDS+69           ;表示月データの登録
        CALL  !NDSCHV          ;表示データを数値に変換
        MODE3=#2
        SETI  MD_CG3F          ;モード変更状態に設定
    ends
else
    B=CUR_ADR (A)              ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)          ;数値データを設定
    if(CUR_ADR<#70)           ;設定データ最下位桁以内
        CUR_ADR++
    endif
endif
ENDIF
RET

;*****
;      休日設定処理
;*****
HOL_D:
    if_bit(MD_CG3F)
        CLRI  MD_CG3F

        HL=#LCDDS+90
        DE=#DAY_D              ;LCD4行目のデータを設定
        CALL  !DT1LSET
        CUR_ADR=#99
        SETI  LCDCHGF
    endif

    IF_BIT(KY_SIGF)
        CLRI  KY_SIGF
        IF(KEY_CD<#'0' || KEY_CD>#'9') ;数字コード以外?
            switch(KEY_CD)         ;キー・コード分岐
            case CLEAR:
                MODE1=#0           ;前面モードに変更
                SETI  MD_CG1F
                break
            case DELETE:
                HL=#LCDDS+99
                CALL  !DDSCNV        ;表示データを数値に変換
                H=#ADRSTA
                L=#00H
                M_BANK=#0
                CALL  !BANKSET
                B=#HOL_DAT          ;休日データの検索
                CALL  !HOLSEA        ;検索アドレス決定
                if_bit(ADSEAF)
                    CALL  !DEL_IL    ;データ削除
                    HOL_VOL--        ;休日データ数をデクリメント
                endif
                MODE1=#0
                SETI  MD_CG1F
                break
            case CUR_RI:
                if(CUR_ADR<#100)    ;日データ2桁目入力
                    CUR_ADR++      ;カーソル・アドレスに1加算
                endif
                break
            case CUR_LE:
                if(CUR_ADR>#99)    ;日データ1桁目入力

```

```

        CUR_ADR--          :カーソル・アドレスから1減算
    endif
    break
case ENTER:              :表示日データの登録
    HL=#LCDDS+99
    CALL !DDSCNV          :表示データを数値に変換
    H=#ADRSTA
    L=#00H
    M_BANK=#0
    CALL !BANKSET
    B=#HOL_DAT           :休日データの検索
    CLR! ADENDF
    while_bit(!ADENDF)   :検索アドレス決定
        CALL !ADR_SEA
        if_bit(!CY)
            CALL !BANKSET :バンク変更
        endif
    endwhile
    MEMPAG=H (A)         :検索アドレスの登録
    MEMBNK=M_BANK (A)   :検索メモリ・バンクの登録
    if([HL]=B) (A)      :休日データあり
        A=HOL_VOL
        A&=#0FH         :新規ページに登録(データ数=X0H)
        if_bit(Z)
            CALL !HOL_SAV :休日データ登録
        else
            A=HOL_VOL   :休日データ値の設定
            X=#10H      :アドレス値の設定
            AX=X
            A+=MEMPAG   :検索アドレス値の加算
            HL=AX
            if(H>#ADREND) (A) :最終アドレス以上
                A=H
                A-=#ADREND :次バンクへのアドレス補正
                H=A
                M_BANK++   :次バンクに変更
                CALL !BANKSET
            endif
            [HL]=#HOL_DAT (A) :データ種類の設定
            HL++
            [HL]=CE_YEAD (A)  :年データ設定
            HL++
            [HL]=CE_MOND (A) :月データの設定
            HL++
            [HL]=CE_DAYD (A) :日データの設定
            HL++
            A=#0
            for(i=#4;i<#16;i++) :4-15バイトのデータをクリア
                [HL]=A
                HL++
            next
            HOL_VOL++
            NODE1=#0        :登録終了
            SET! MD_CG!F
        endif
    else
        CALL !HOL_SAV
    endif
ends
ELSE
    B=CUR_ADR (A)         :カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)     :数値データを設定
    if(CUR_ADR<#100)     :設定データ最下位桁以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

;*****
;   メモリ・オーバーフロー処理
;*****
HOL_OV:
    if_bit(KY_SIGF)

```

```

        CLR1    KY_SIGF
        switch(KEY_CD)
        case CLEAR:                ;前モードに設定
            MODE2=#0
            SET1    MD_CG2F
        ends
    endif
    RET

;*****
;    休日データ登録処理
;*****
HOL_SAV:
    M_BANK=#3                    ;空き領域あり
    CALL    !BANKSET
    A=!0BF00H
    if(A=#VAC_DAT)              ;最終アドレスのメモリ空きあり
        M_BANK=MEMBNK (A)       ;外部メモリを開始位置に設定
        CALL    !BANKSET
        X=HOL_VOL (A)           ;休日データ値の設定
        C=#10H                  ;アドレス値の設定
        A=#0
        AX/=C
        A=X
        A+=MEMPAG               ;検索アドレス値の加算
        H=A
        L=#0
        if(H>#ADREND) (A)       ;最終アドレス以上
            A=H
            A-=#ADREND          ;次バンクへのアドレス補正
            H=A
            M_BANK++           ;次バンクに変更
            CALL    !BANKSET
        endif
        DE=#DTWORK              ;転送用データの作成
        [DE]=#HOL_DAT (A)       ;データ種類の設定
        DE++
        [DE]=CE_YEAD (A)        ;年データ設定
        DE++
        [DE]=CE_MOND (A)       ;月データの設定
        DE++
        [DE]=CE_DAYD (A)       ;日データの設定
        DE++
        A=#0
        for(i=#4;i<#16;i++)     ;4-15バイトのデータをクリア
            [DE]=A
            DE++
        next
        A=#VAC_DAT              ;LCD表示データをワーク・エリアに設定
        for(i=#0;i<#240;i++)
            [DE]=A
            DE++
        next
        DE=#DTWORK
        L=#0
        CALL    !DT_SAV         ;データ登録
        HOL_VOL++
        MODE1=#0                ;登録終了
        SET1    MD_CG1F
    else
        SET1    MD_CG3F        ;メモリ・アドレスに空きなし
        MODE3=#3                ;オーバーフロー・モードに設定
    endif
    RET

;*****
;    修正モード・データ登録処理
;*****
CAL_CSV:
    M_BANK=MEMBNK (A)
    CALL    !BANKSET
    H=MEMPAG (A)                ;登録アドレスの設定
    L=#16
    DE=#LCDDS
    CALL    !DTIPSET            ;LCD表示データの登録

```



```

RET

;*****
;   スケジュール・タイトル表示
;*****
SCA_TID:
CALL  !LCD_DSC
HL=#LCDDS                               ;日付設定先頭アドレス
X=CE_YEAD (A)                            ;年データの取り込み
CALL  !YEADSP                             ;年データ表示
[HL]=#DIV_SIG (A)                         ;/
HL++
X=CE_MOND (A)                             ;月データの取り込み
CALL  !MONDSP                             ;月データ表示
[HL]=#DIV_SIG (A)                         ;/
HL++
X=CE_DAYD (A)                             ;日データの取り込み
CALL  !MONDSP                             ;日データ表示
HL=#LCDDS+30                             ;2行目に表示
DE=#REG_D
CALL  !DTILSET
HL=#LCDDS+210                             ;8行目に表示
DE=#DEL_D
CALL  !DTILSET
RET

;*****
;   カレンダー月変更処理
;*****
BEF_MON:
if(CE_MOND>#1)
  CE_MOND--                               ;前の月に設定
  SETI  CLDCGF
elseif(CE_YEAD>#1)
  CE_YEAD--                               ;1901年以降
  CE_MOND=#12                             ;12月に設定
  SETI  CLDCGF
endif
if_bit(CLDCGF)
  SETI  PAR_1F                             ;()を1日に設定
  CLR1  PAR_SET
endif
RET

NEX_MON:
if(CE_MOND<#12)
  CE_MOND++                               ;次の月に設定
  SETI  CLDCGF
elseif(CE_YEAD<#199)
  CE_YEAD++                               ;2099年以降
  CE_MOND=#1                             ;1月に設定
  SETI  CLDCGF
endif
if_bit(CLDCGF)
  SETI  PAR_1F                             ;()を1日に設定
  CLR1  PAR_SET
endif
RET

;*****
;   カレンダー年変更処理
;*****
BEF_YEA:
if(CE_YEAD>#1)
  CE_YEAD--                               ;1901年以降
  SETI  CLDCGF                             ;前の年に設定
  CLR1  PAR_SET
endif
RET

NEX_YEA:
if(CE_YEAD<#199)
  CE_YEAD++                               ;2099年以降
  SETI  CLDCGF                             ;次の年に設定
  CLR1  PAR_SET
endif

```

```

endif
RET

;*****
; カレンダ年表示処理
;*****
YEADSP:
PUSH BC
if(X<#100) (A) ;19XX年?
[HL]=#'1' (A) ;19表示の設定
HL++
[HL]=#'9' (A)
HL++
elseif(X<#200) (A) ;20XX年?
[HL]=#'2' (A) ;20表示のデータ
HL++
[HL]=#'0' (A)
HL++
A=X ;年データを20XX年用に補正(100減算)
A-=#100
X=A ;年データの設定
else
[HL]=#'2' (A) ;21表示のデータ
HL++
[HL]=#'1' (A)
HL++
A=X ;年データを21XX年用に補正(100減算)
A-=#200
X=A ;年データの設定
endif
A=#0
C=#10 ;データ10進補正
AX/=C ;除算結果X:十の桁, C:一の桁
A=X
A+#30H ;表示コードに変換
[HL]=A ;十の桁設定
HL++
A+#30H
A+=C
[HL]=A ;一の桁設定
HL++
POP BC
RET

;*****
; カレンダ年表示→数値変換処理
;*****
YDSCNV:
A=[HL+2] ;年(十の桁)データの取り込み
A&=#0FH ;数値データに変換
X=#10 ;10進補正
AX*=X
A=[HL+3] ;年(一の桁)データの取り込み
A&=#0FH ;数値データに変換
X+=A ;下位2桁の年データ設定
if([HL]==#'1') (A) ;19XX年
if(XI=#0) (A) ;加算値=0
A=#0
else
AX=#1 ;入力=1X00のときは1901年に設定
endif
else
A=#100 ;加算値=100(20XX年)
endif
A+=X
CE_YEAD=A ;表示年データの設定
RET

;*****
; カレンダ月表示→数値変換処理
;*****
MDSNV:
if([HL+1]!="#SPC") (A) ;2桁目入力あり
if([HL]==#'1') (A) ;1桁目入力あり
A=#10 ;10を加算

```

```

else
    A=#0 ; 0 を加算
endif
A+=[HL+1] ; 2 桁目のデータを加算
A-=#30H
else
    if([HL]!="#SPC) (A) ; 1 桁目入力あり
        A=[HL] ; 1 桁目のデータを加算
        A-=#30H
    else
        A=#1 ; 入力なしのときは 1 月に設定
    endif
endif
if(A==#0) ; データが 0 のとき
    A=#1 ; 1 月に設定
endif
if(A>#12) ; エラー補正
    A=#12 ; 1 2 月に設定
endif
CE_MOND=A
RET

;*****
; カレンダー日表示→数値変換処理
;*****
DDSCNV:
    if([HL+1]!="#SPC) (A) ; 2 桁目入力あり
        if([HL]<#'4') (A) ; 1 桁目入力あり
            A=[HL]
            A&=#03H ; 上位桁を加算
            X=#10 ; 入力数値を 10 倍
            AX*=-X
            A=X
        else
            A=#0 ; 0 を加算
        endif
        A+=[HL+1] ; 2 桁目のデータを加算
        A-=#30H
    else
        if([HL]!="#SPC) (A) ; 1 桁目入力あり
            A=[HL] ; 1 桁目のデータを加算
            A&=#0FH
        else
            A=#1 ; 入力なしのときは 1 日に設定
        endif
    endif
    C=A

    A=YEAD ; 現在月の最終日を参照
    A&=#0000011B
    if(A==#0) ; うるう年 (下位 2 ビット = 0) ?
        HL=#CAL_LEA ; うるう年データの参照
    else
        HL=#CAL_NOR ; 平年データの参照
    endif
    B=MOND (A)
    B--
    A=[HL+B] ; 表示日データの設定
    if(A)=C ; 設定日が最終日以内
        A=C ; 最終日を設定
    endif
    if(A==#0) ; データが 0 のとき
        A=#1 ; 1 日に設定
    endif
    CE_DAYD=A
    RET

;*****
; カレンダー月/日表示処理
;*****
MONDSP:
    PUSH BC
    A=#0
    C=#10

```

```

AX/=C                                ;除算結果X：十の桁，C：一の桁
if(X==#0) (A)                        ;10月以降
  A=#SPC                              ;十の桁に空白データを設定
else
  A=X                                  ;十の桁に1データを設定
  A+=#30H
endif
[HL]=A
HL++
A=C
A+=#30H                              ;データ変換
[HL]=A                                ;一の桁に月データ設定
HL++
POP BC
RET

;*****
; 日付検索 (アドレス前通)
;*****
DATE_F:
CLR1 ADSEAF                          ;データ検索
while_bit(!ADSEAF)
  CALL !ADR_SEA
  if_bit(CY)                          ;バンク変更
  if([HL]==B) (A)                    ;検索データの種類一致
    if([HL+1]==CE_YEAD) (A)          ;年データ一致
      if([HL+2]==CE_MOND) (A)       ;月データ一致
        if([HL+3]==CE_DAYD) (A)    ;日データ一致
          SET1 CY
          SET1 ADSEAF
          MEMPAG=H (A)              ;検索アドレスをメモリに格納
          MEMBNK=M_BANK (A)
        else
          CALL !ADR_INC
          if_bit(ICY)
          CALL !BANKSET
        endif
      endif
    else
      CALL !ADR_INC
      if_bit(ICY)
      CALL !BANKSET
    endif
  endif
else
  CALL !ADR_INC
  if_bit(ICY)
  CALL !BANKSET
endif
endif
else
  break
endif
else
  if(M_BANK<=#3)
    CALL !BANKSET                  ;バンク変更
  else
    break
  endif
endif
endw
RET

;*****
; 日付検索 (アドレス後通)
;*****
DATE_R:
CLR1 ADSEAF                          ;データ検索
while_bit(!ADSEAF)
  CALL !AD_SEAR
  if_bit(CY)                          ;バンク変更
  if([HL]==B) (A)                    ;検索データの種類一致
    if([HL+1]==CE_YEAD) (A)          ;年データ一致
      if([HL+2]==CE_MOND) (A)       ;月データ一致
        if([HL+3]==CE_DAYD) (A)    ;日データ一致

```

```

        SETI    CY
        SETI    ADSEAF
        MENPAG=H (A)      ;検索アドレスをメモリに格納
        MEMBNK=M_BANK (A)
    else
        CALL    !ADR_DEC
        if_bit(!CY)
        CALL    !BANKSET
    endif
endif
else
    CALL    !ADR_DEC
    if_bit(!CY)
    CALL    !BANKSET
endif
endif
else
    CALL    !ADR_DEC
    if_bit(!CY)
    CALL    !BANKSET
endif
endif
else
    break
endif
else
    if(M_BANK!=#0)
        CALL    !BANKSET      ;バンク変更
    else
        break
    endif
endif
endw
RET

```

```

;*****
;      休日検索
;*****
HOLSEA:

```

```

    CLRl    ADSEAF
    while_bit(!ADSEAF)      ;データ検索
        CALL    !ADR_SEA
        if_bit(CY)          ;バンク変更
            if([HL]=B) (A)   ;検索データの種類一致
                if([HL+1]=CE_YEAD) (A) ;年データ一致
                    if([HL+2]=CE_MOND) (A) ;月データ一致
                        if([HL+3]=CE_DAYD) (A) ;日データ一致
                            SETI    CY
                            SETI    ADSEAF
                            MENPAG=H (A) ;検索アドレスをメモリに格納
                            MEMBNK=M_BANK (A)
                        else
                            A=L
                            A+=#10H
                            L=A
                            if(A==#0)
                                CALL    !ADR_INC
                                if_bit(!CY)
                                CALL    !BANKSET
                            endif
                        endif
                    endif
                else
                    A=L
                    A+=#10H
                    L=A
                    if(A==#0)
                        CALL    !ADR_INC
                        if_bit(!CY)
                        CALL    !BANKSET
                    endif
                endif
            endif
        else
            A=L
            A+=#10H
            L=A
            if(A==#0)
                CALL    !ADR_INC
                if_bit(!CY)
                CALL    !BANKSET
            endif
        endif
    endwhile

```



```

        A=L
        A+=#10H
        L=A
        if(A=#0)
            CALL IADR_INC
            if_bit(!CY)
            CALL !BANKSET
        endif
    endif
endif
else
    break
endif
else
    if(M_BANK<=#3)
        CALL !BANKSET ;バンク変更
    else
        break
    endif
endif
endw
RET

;*****
; スケジュール表示
;*****
SCA_DSP:
    CALL !LCD_DSC
    HL=#LCDDS ;日付設定先頭アドレス
    X=CE_YEAD (A) ;年データの取り込み
    CALL !YEADSP ;年データ表示
    [HL]=#DIV_SIG (A) ;/
    HL++
    X=CE_MOND (A) ;月データの取り込み
    CALL !MONDSP ;月データ表示
    [HL]=#DIV_SIG (A) ;/
    HL++
    X=CE_DAYD (A) ;日データの取り込み
    CALL !MONDSP ;日データ表示
    HL=#LCDDS+30 ;カーソル・アドレスの設定
    DE=#REG_D
    CALL !DTILSET ;0000H+30(2行目)に設定
    HL=#LCDDS+210 ;カーソル・アドレスの設定
    DE=#DEL_D
    CALL !DTILSET ;0000H+210(8行目)に設定
    SETI IV_CHGF ;反転表示行を2行目に設定
    INV_DS=#2
    CLRI CR_DSPF
    SETI LCDCHGF
    RET

;*****
; カレンダー算出用データ
;*****
CALE_T: ;曜日移動データ(上位:うるう年, 下位:平年)
    DB 00H ;1月
    DB 33H ;2月
    DB 43H ;3月
    DB 06H ;4月
    DB 21H ;5月
    DB 54H ;6月
    DB 06H ;7月
    DB 32H ;8月
    DB 65H ;9月
    DB 10H ;10月
    DB 43H ;11月
    DB 65H ;12月

CAL_LEA: ;月日数データ(うるう年)
    DB 31 ;1月
    DB 29 ;2月
    DB 31 ;3月
    DB 30 ;4月
    DB 31 ;5月
    DB 30 ;6月
    
```

```

DB      31                ; 7月
DB      31                ; 8月
DB      30                ; 9月
DB      31                ; 10月
DB      30                ; 11月
DB      31                ; 12月
    
```

CAL_NOR: ;月日数データ(平年)

```

DB      31                ; 1月
DB      28                ; 2月
DB      31                ; 3月
DB      30                ; 4月
DB      31                ; 5月
DB      30                ; 6月
DB      31                ; 7月
DB      31                ; 8月
DB      30                ; 9月
DB      31                ; 10月
DB      30                ; 11月
DB      31                ; 12月
    
```

```

;*****
; カレンダー1箇月表示データ
;*****
    
```

CALDSP:

```

DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, '1', SPC, '2', SPC, '3', SPC, '4', SPC, '5'
DB      SPC, '6', SPC, '7', SPC, '8', SPC, '9', '1', '0'
DB      '1', '1', '1', '2', '1', '3', '1', '4', '1', '5'
DB      '1', '6', '1', '7', '1', '8', '1', '9', '2', '0'
DB      '2', '1', '2', '2', '2', '3', '2', '4', '2', '5'
DB      '2', '6', '2', '7', '2', '8', '2', '9', '3', '0', '3', '1'
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
END
    
```

4.7 メモ帳モード

メモ帳モードでは、モード選択や、データの登録、修正、検索、削除を行っています。

4.7.1 登録データのフォーマット

メモ帳データの登録フォーマットは、タイトル(レベル1)用と、タイトル(レベル2)の表示データ用の2種類があります。各登録フォーマットを図4-15、図4-16に示します。

図4-15 タイトル(レベル1)の登録フォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xx00H	06	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
xx10H	タイトル(レベル1)										00	00	00	00	00	00
xx20H	タイトル(レベル1)										00	00	00	00	00	00
⋮																
xxE0H	タイトル(レベル1)										00	00	00	00	00	00
xxF0H	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

図4-16 メモ帳データの登録フォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xx00H	03	タイトル(レベル1)										00	00	00	00	00
xx10H	データの種類 データ検索用パラメータ LCD表示データ240バイト (30文字×8行)															
xx20H																
xx30H																
xx40H																
xx50H																
xx60H																
xx70H																
xx80H																
xx90H																
xxA0H																
xxB0H																
xxC0H																
xxD0H																
xxE0H																
xxF0H																

4.7.2 モード選択条件

メモ帳モードの遷移状態の条件を表4 - 8に示します。

表4 - 8 メモ帳モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
4	0	-	-	-	メモ帳モード選択
	1	0	-	-	タイトル(レベル1)登録モード
		1	-	-	タイトル(レベル2)登録モード
		2	-	-	データ登録モード
		3	-	-	メモリ・オーバフロー表示モード
	2	0	-	-	先頭文字選択モード
		1	0	-	タイトル(レベル1)選択モード
			1	-	タイトル(レベル2)選択モード
		2	-	-	データ修正モード
	3	0	-	-	先頭文字選択モード
		1	0	-	タイトル(レベル1)選択モード
			1	-	タイトル(レベル2)選択モード
		2	-	-	データ表示モード
	4	0	-	-	先頭文字選択モード
		1	0	-	タイトル(レベル1)選択モード
			1	-	タイトル(レベル2)選択モード
		2	-	-	データ表示モード

備考 - : don't care

4.7.3 登録処理

登録モードでは、ENTキー入力によりMODE2の値を更新し、CLRキー入力のときは前モードに戻すためMODE1を0に設定します。登録データには、タイトル(レベル1)と、タイトル(レベル2)と、メモ帳データの3種類があります。

(1) タイトル(レベル1)の登録

タイトル(レベル1)の登録は、文字データのキー入力により、10文字で行います。タイトル(レベル1)の登録の順を次に示します。

データ登録メモリの $\times \times 00\text{H}$ のデータが 06H のアドレスを検索します。

入力されたタイトル(レベル1)と、で検索したデータの $\times \times \times 1\text{H} - \times \times \times \text{AH}$ のデータが一致していないことを確認します。一致するタイトル(レベル1)があるときは、登録処理を行いません。

タイトル(レベル1)の登録には、新規ページに登録するとき(タイトル(レベル1)のデータ数が14で割り切れる)と、登録中のページに登録するときの2種類があります。データ登録後は、データ数(TIT_VOL)を+1します。

登録後は、タイトル(レベル2)の登録モードになります(MODE2=1)。

(a) 新規ページ

タイトル(レベル1)のデータ数が14($\times \times 00\text{H} - \times \times \text{FFH}$ に登録する最大値)で割り切れるときは、メモリ最終ページの空きを確認します。空き領域がないときは、メモリ・オーバフローにします(MODE2=3)。

空き領域があるときは、図4-15に示す $\times \times 00\text{H} - \times \times 1\text{FH}$ のデータを設定し、 $\times \times 20\text{H} - \times \times \text{FFH}$ を空きデータ(FFH)で書き込みます。

(b) 登録中のページ

で検索したアドレスと、タイトル(レベル1)のデータ数(TIT_VOL)から、登録するアドレスを算出し、データを登録します。

(2) タイトル(レベル2)の登録

タイトル(レベル2)の登録は、文字データのキー入力により、30文字で行います。データ設定後のENTキー入力により、タイトル(レベル2)を登録し、メモ入力モードにします(MODE2=2)。

(3) メモ帳の登録

メモ帳登録モードは、CLRキーが入力されたときは、前モードに戻すためMODE1を0に設定します。登録データは、文字データのキー入力により、カーソルのあるアドレス(CUR_ADR)に文字データを書き込みます。ENTキー入力後、次の順で登録処理を行います。

メモリ最終ページの先頭アドレスの確認をします。

バンク3(P61, P60 = 1, 1)のBF00H番地がFFH(空きエリア)以外のときはメモリ・オーバーフローにします(MODE2 = 3に設定)。

データを登録するアドレスを検索します。

検索条件: $\times \times 00H$ が04H以上、または、 $\times \times 00H$ が03Hで入力したタイトル(レベル2)のデータが、表示コードで比較して小さい値のとき

検索したアドレスにデータを登録します。表示データ(240文字)は、 $\times \times 10H - \times \times FFH$ に登録し、 $\times \times 00H$ に03Hを、 $\times \times 01H - \times \times 0AH$ にタイトル(レベル1)を、 $\times \times 0BH - \times \times 0FH$ に00Hを登録します。検索アドレス以降のデータは、次のアドレス(+100H)にシフトします。

データ数(MEM_VOL)を+1します。

4.7.4 データ選択処理

データ選択処理では、先頭文字選択モード(MODE2 = 0)と、タイトル選択モード(MODE2 = 1)の2種類があります。

(1) 先頭文字選択モード

先頭文字選択モードでは、
、
、
キー入力により、カーソルのアドレス(CUR_ADR)の値に加算、減算をしてカーソルの位置を変更しています。ENTキー入力により、先頭文字を決定し(SEACHRにA-Zを設定)、タイトル(レベル1)選択モードに設定します(MODE2 = 1)。ただし、NEXT PAGEを選択したときは、データ登録メモリの $\times \times 00H$ に03Hが格納されている先頭アドレスを検索し、次のモード(MODE2 = 3)に移行します。検索データがないときは、メモ帳モード選択に戻ります(MODE1 = 0)。

図4-17 メモ帳データの検索

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\times \times 00H$	03	タイトル(レベル1)										00	00	00	00	00

03Hで格納されている先頭を検索

(2) タイトル選択モード

タイトル選択では、レベル1とレベル2の2段階で行います。

(a) タイトル(レベル1)の場合

タイトル(レベル1)のときは、データ登録メモリの $x \times 00H = 03H$ で $x \times 10H - x \times E0H$ の $x \times 0H = SEACHR$ のタイトル(レベル1)を検索して表示します。検索データは、1回の処理で7個まで表示します。表示しているタイトル(レベル1)の先頭/最終アドレスをメモリ(REV_PNT/FOR_PNT)に格納し、次の検索時の開始アドレスとして使用します。

UP/DOWNキーにより、検索文字のデータ(SEACHR)を+1/-1しています。検索文字変更後は、データ登録メモリの先頭から開始します。

ENTキー入力により、反転表示しているタイトル(レベル1)を決定し、タイトル(レベル2)選択モードにします(MODE2=2)。

(b) タイトル(レベル2)の場合

タイトル(レベル2)のときは、データ登録メモリの $x \times 00H = 03H$ で、 $x \times 01H - x \times 0AH$ のデータとタイトル(レベル1)のデータが一致しているアドレスを検索して表示します。検索データは、1回の処理で7個まで表示します。表示している名前の先頭/最終アドレスをメモリ(REV_ADR/FOR_ADR)に格納し、次の検索時の開始アドレスとして使用します。

ENTキー入力により、反転表示しているタイトル(レベル2)と $x \times 10H$ から30文字データが一致しているアドレスを検索し、データ表示のモードに移行します(MODE2=3)。

図4-18 タイトル(レベル2)の検索

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$x \times 00H$	02	タイトル(レベル1)										00	00	00	00	00
$x \times 10H$	x	x	タイトル(レベル2)													
$x \times 20H$	SEACHR										(30文字)					

4.7.5 修正処理

修正モードでは、ENTキー入力により修正完了とし、修正前のデータ登録メモリのアドレスに再登録します。登録後は、メモ帳モード選択に戻します(MODE1=0)。CLRキー入力ときは、前モードに戻すためMODE1を0に設定します。修正は、文字データのキー入力により、カーソルのあるアドレス(CUR_ADR)に文字データを上書きしています。

修正データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの+100H/-100Hの $x \times 00H$ データが03Hのときは、表示データの変更をします。

4.7.6 検索処理

検索モードでは、検索データ表示中にUP/DOWNキーを入力することで、表示中のデータ登録メモリの + 100H / - 100H の $\times \times 00H$ データが 03H のときは、表示データの変更をします。

4.7.7 削除処理

削除モードでは、DELキー入力により表示中のデータを削除します。データ削除後は、メモ帳モード選択に戻します (MODE1 = 0)。CLRキー入力のときは、前モードに戻すためMODE1を0に設定します。

削除データ表示中のUP/DOWNキーにより、表示中のデータ登録メモリの + 100H / - 100H の $\times \times 00H$ データが 03H のときは、表示データの変更をします。次の順でデータ削除を行っています。

削除データのアドレスを検索します。

検索したアドレスを、次のアドレス (+ 100H) データから最終アドレス (P61, P60 = 1, 1, BFFFH) までシフトし、上書きすることでデータを削除します。

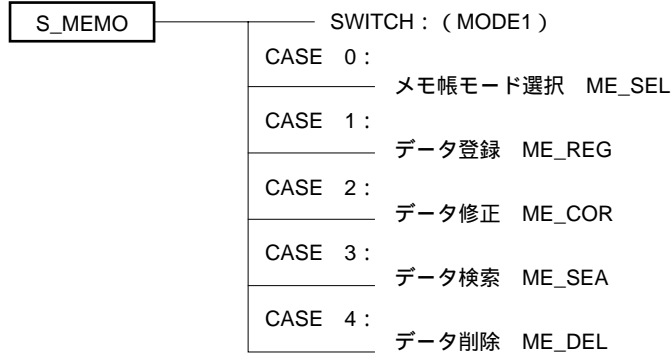
データ数 (MEM_VOL) を - 1 します。

最終アドレスに空きデータ (FFH) を書き込みます。

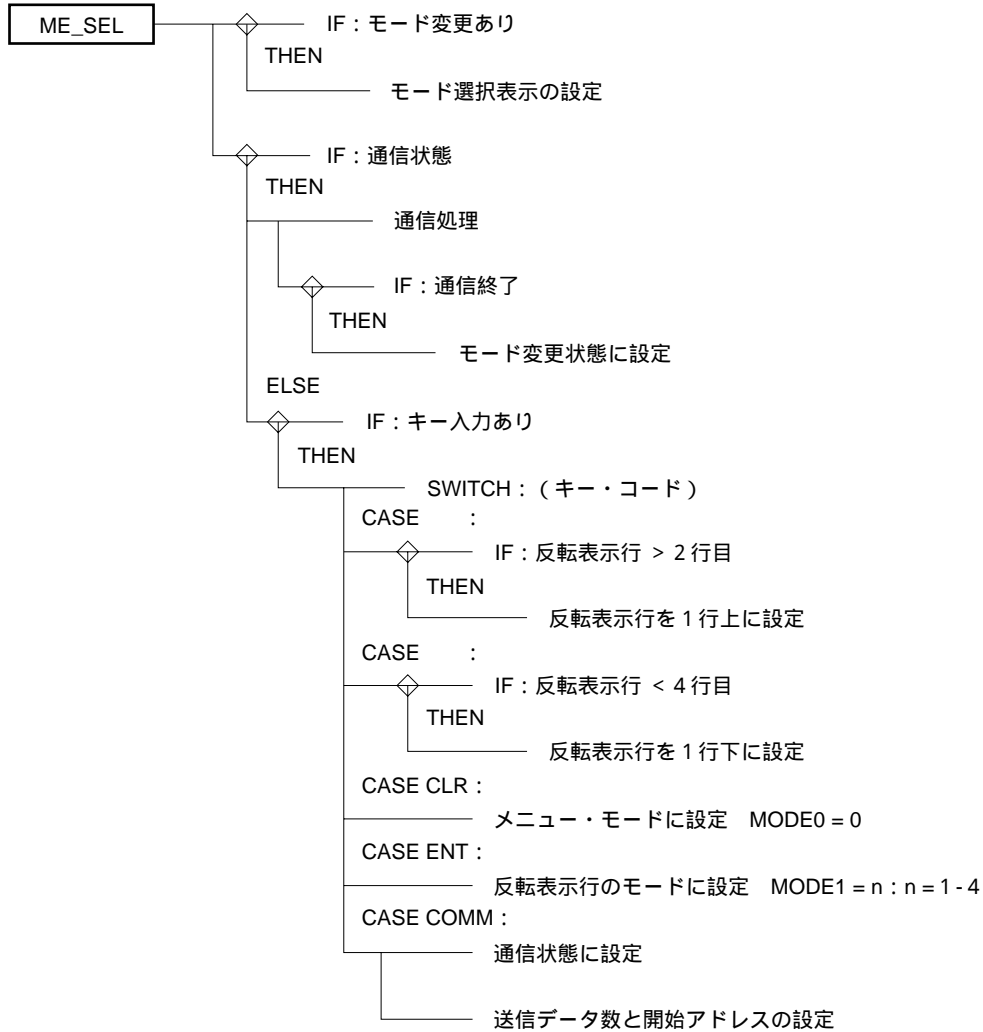
また、タイトル (レベル1) 選択モードのときに、DELキーを入力すると $\times \times 01H$ - $\times \times 0AH$ にタイトル (レベル1) と一致するデータがある間、 - の動作を繰り返します。

4.7.8 SPDチャート

メモ帳モード

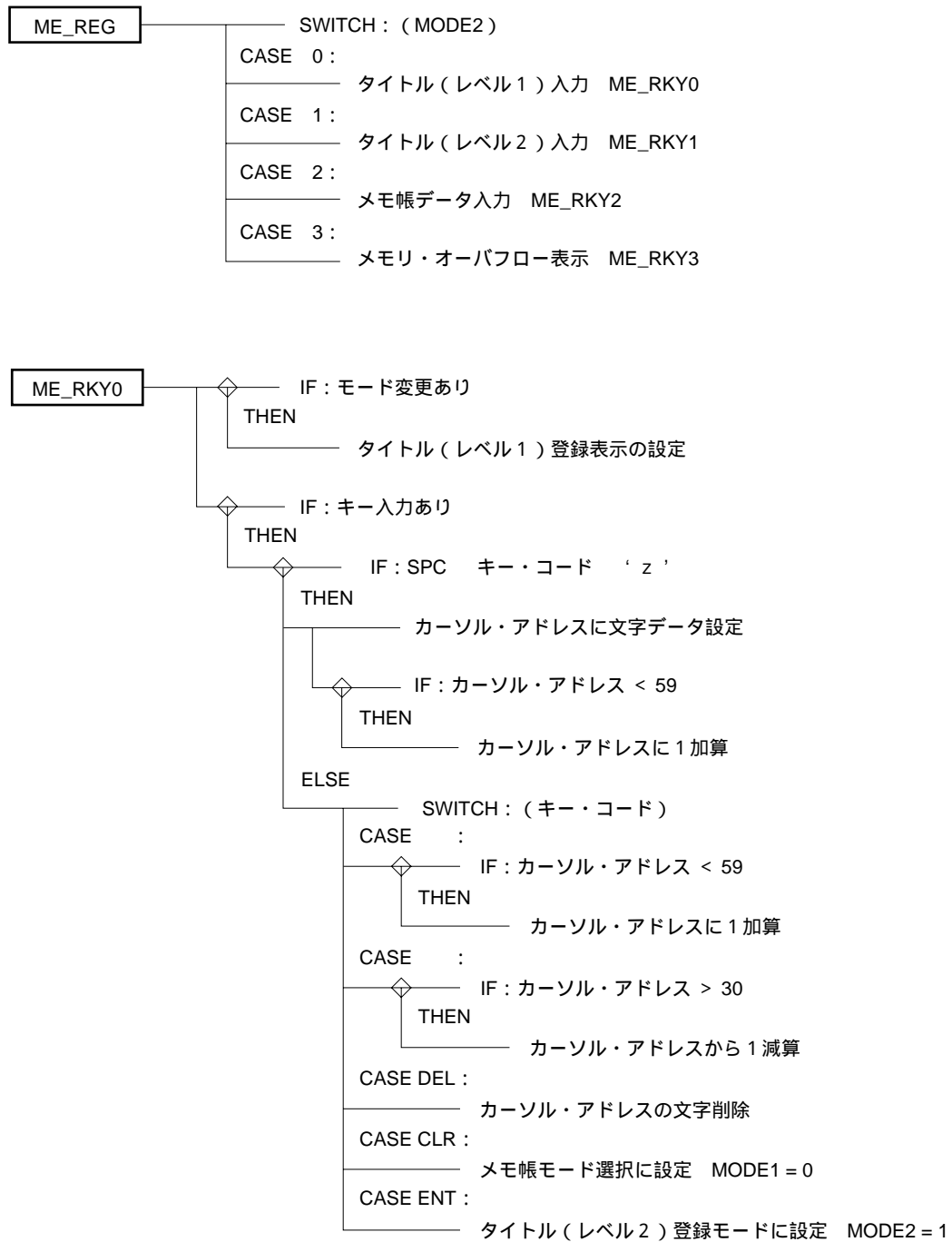


・モード選択



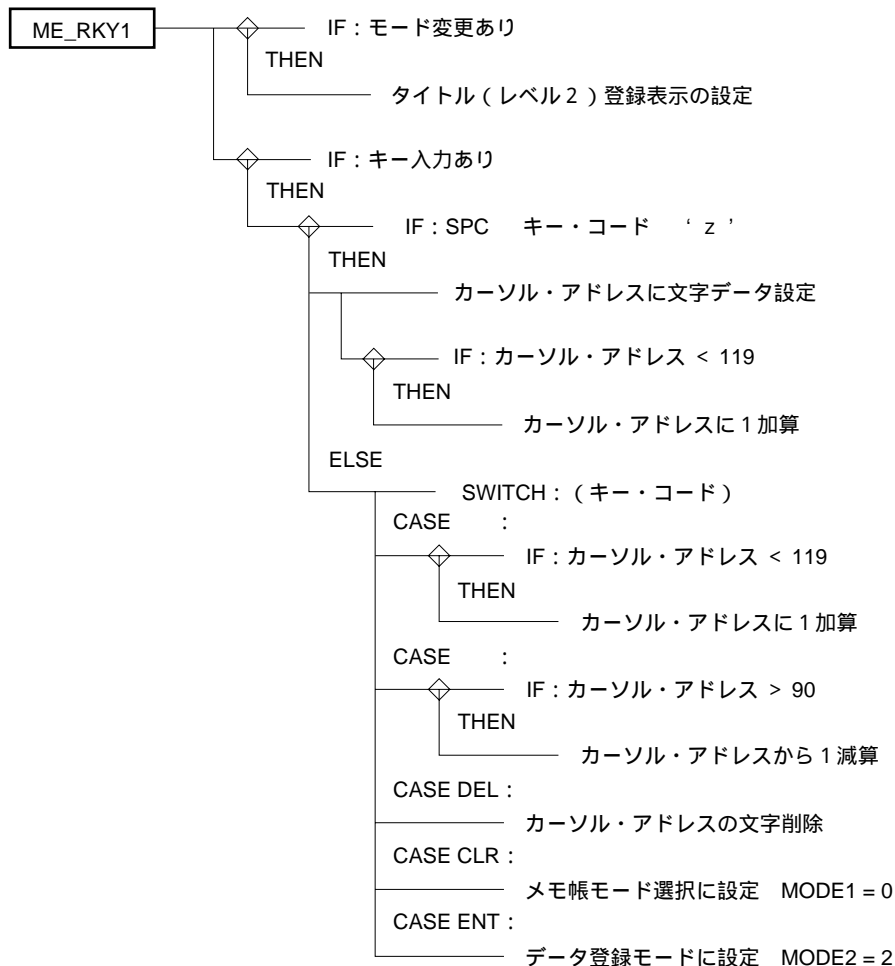
メモ帳モード

・登録モード



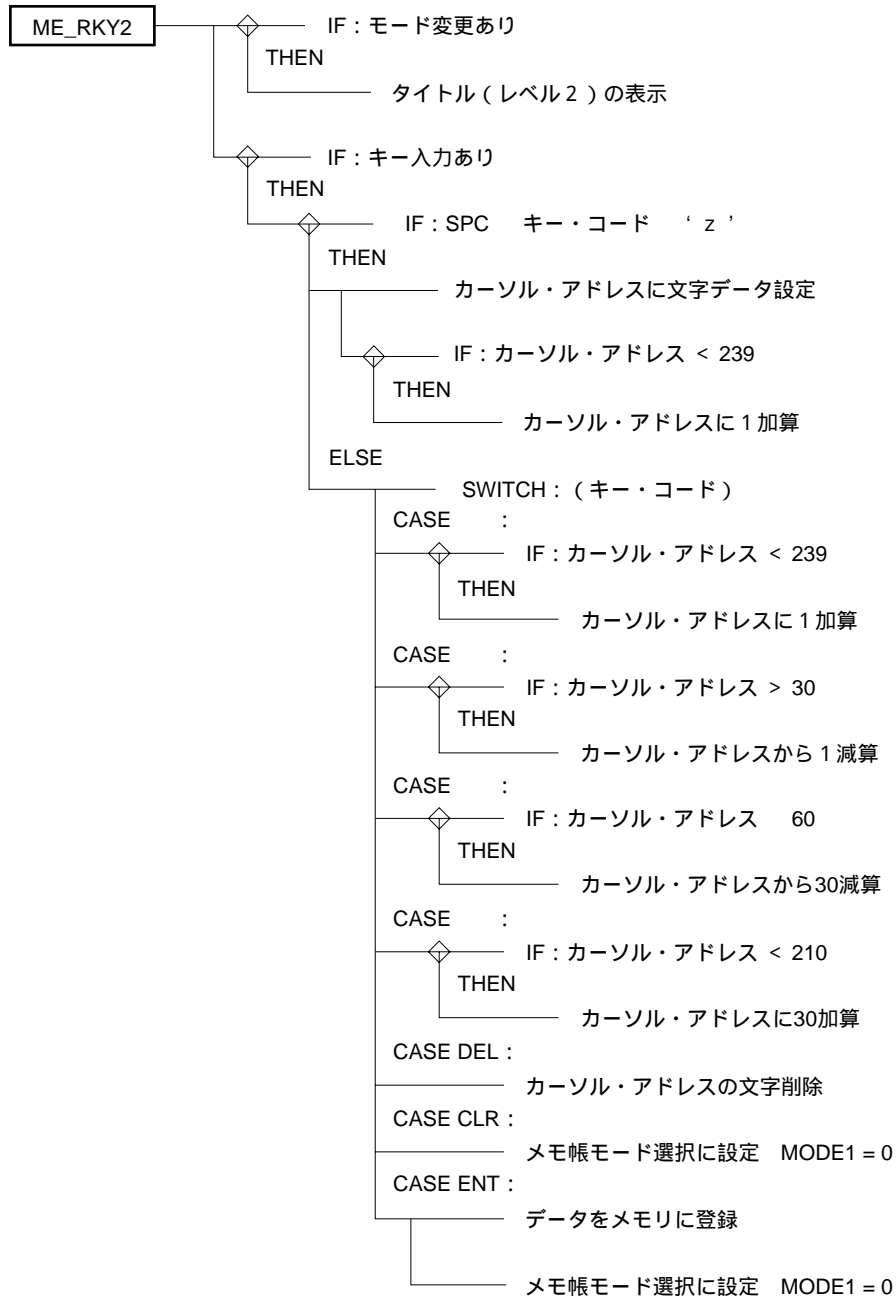
メモ帳モード

・登録モード



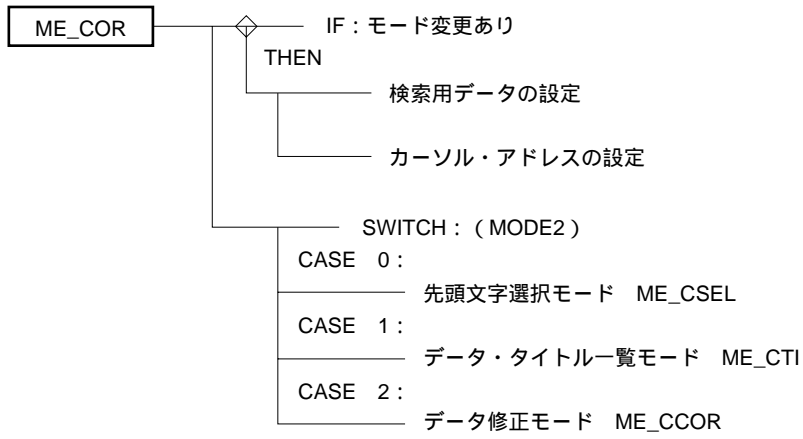
メモ帳モード

・登録モード



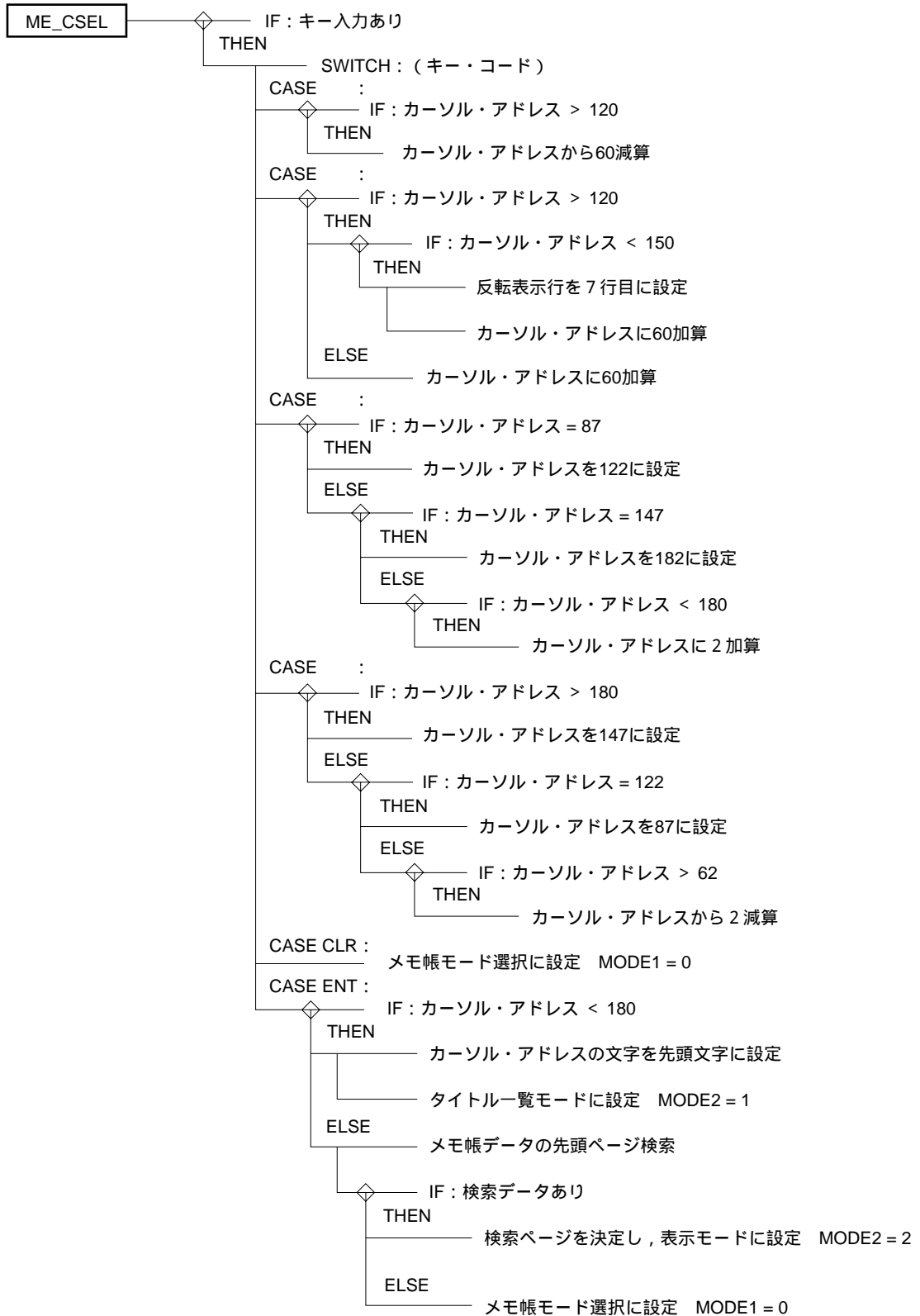
メモ帳モード

・修正モード



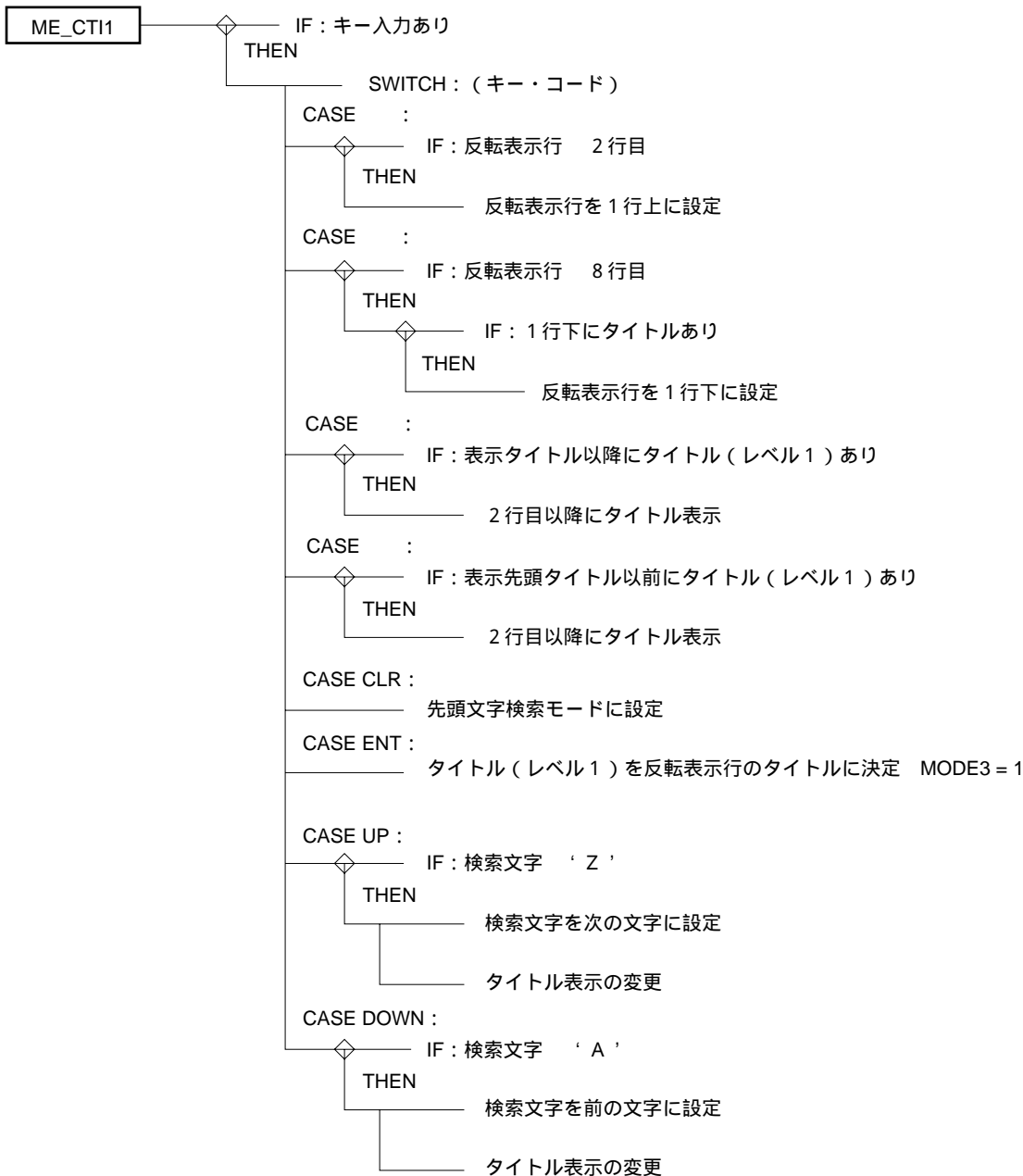
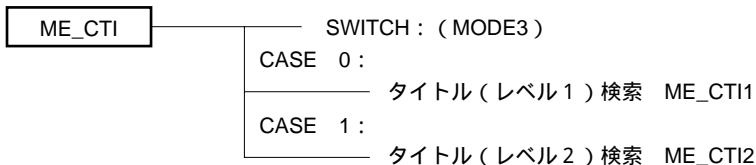
メモ帳モード

・修正モード



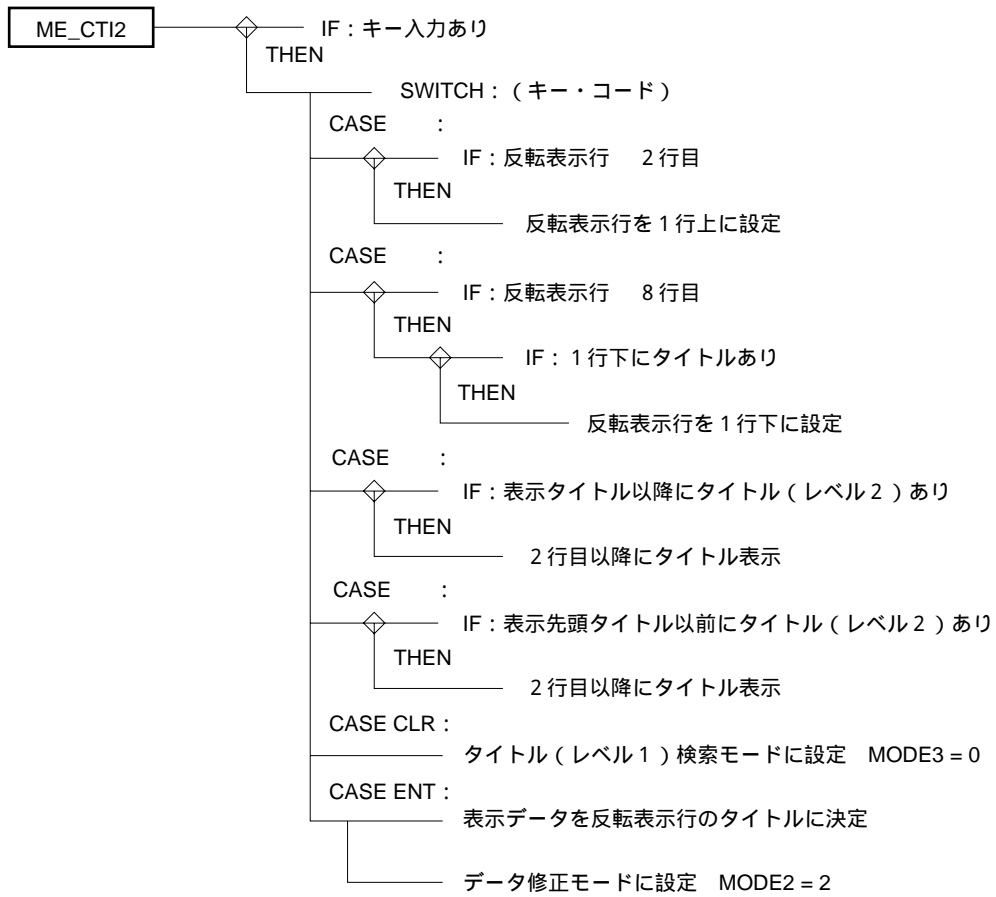
メモ帳モード

・修正モード



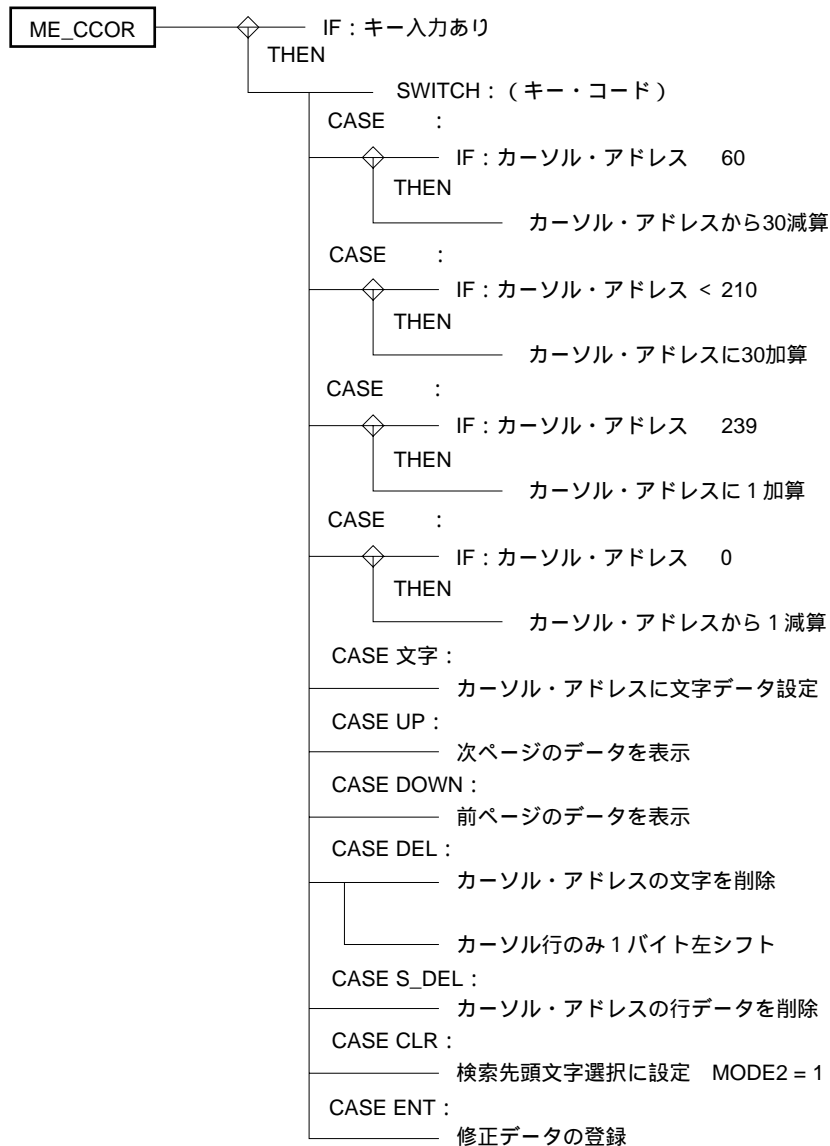
メモ帳モード

・修正モード



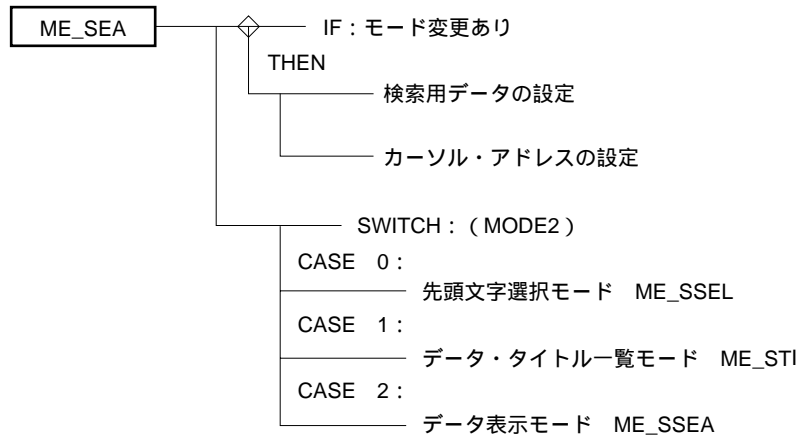
メモ帳モード

・修正モード



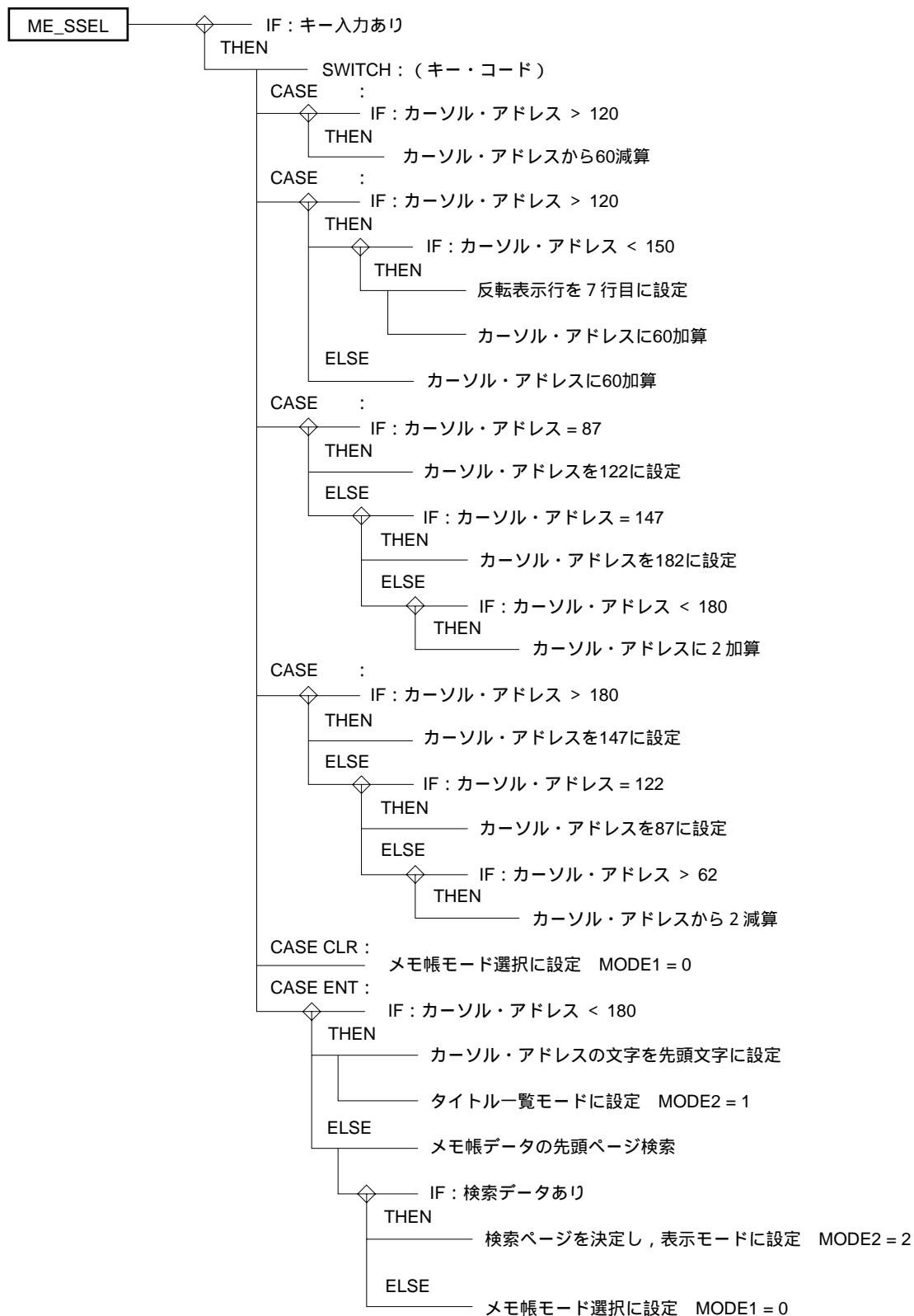
メモ帳モード

・検索モード



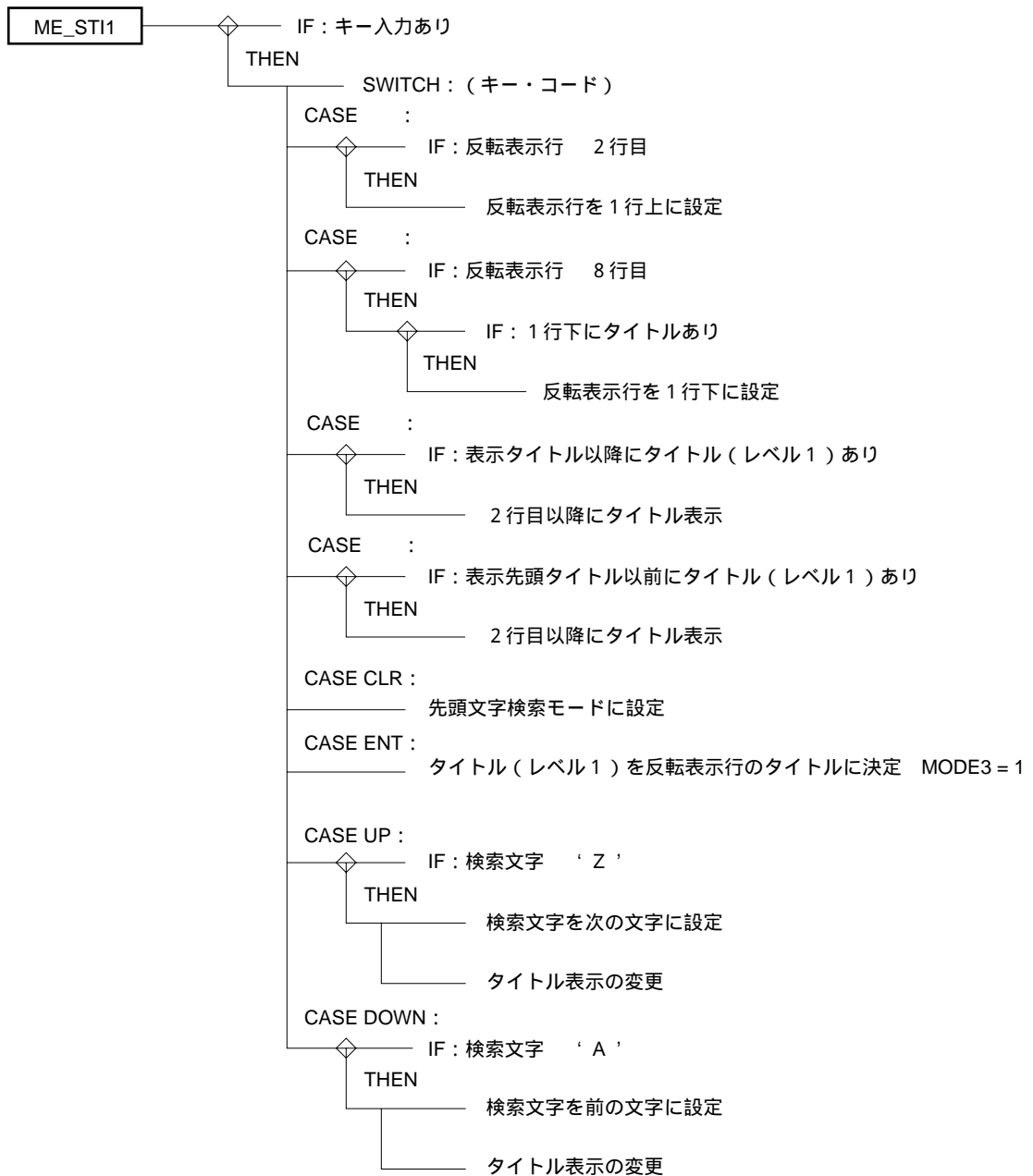
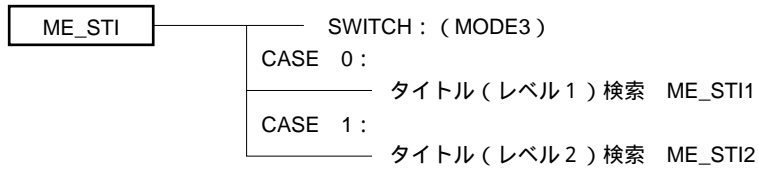
メモ帳モード

・検索モード



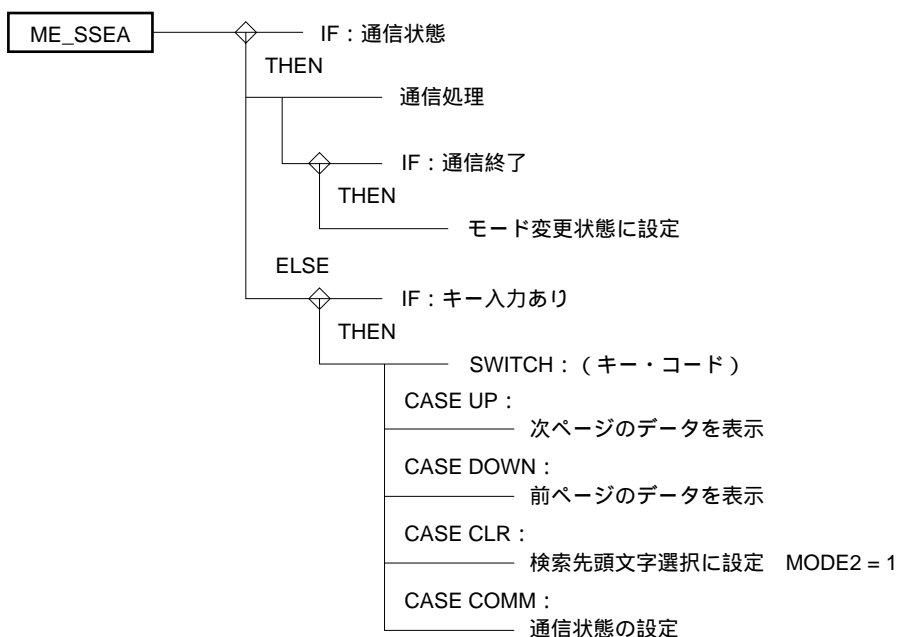
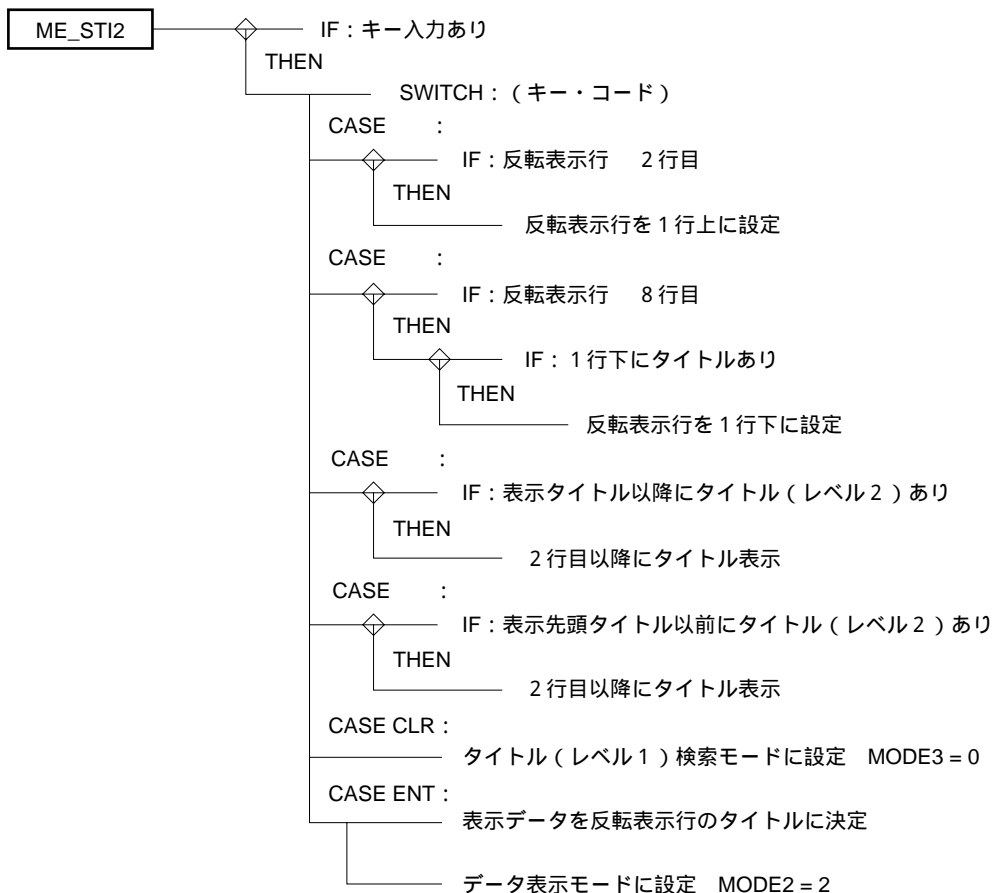
メモ帳モード

・検索モード



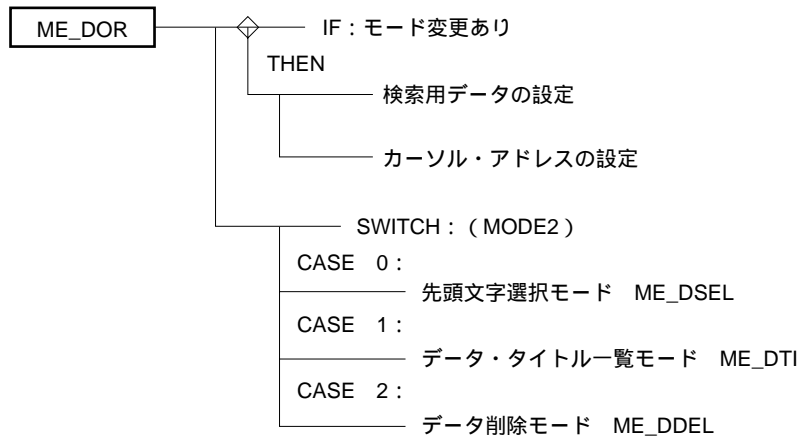
メモ帳モード

・検索モード



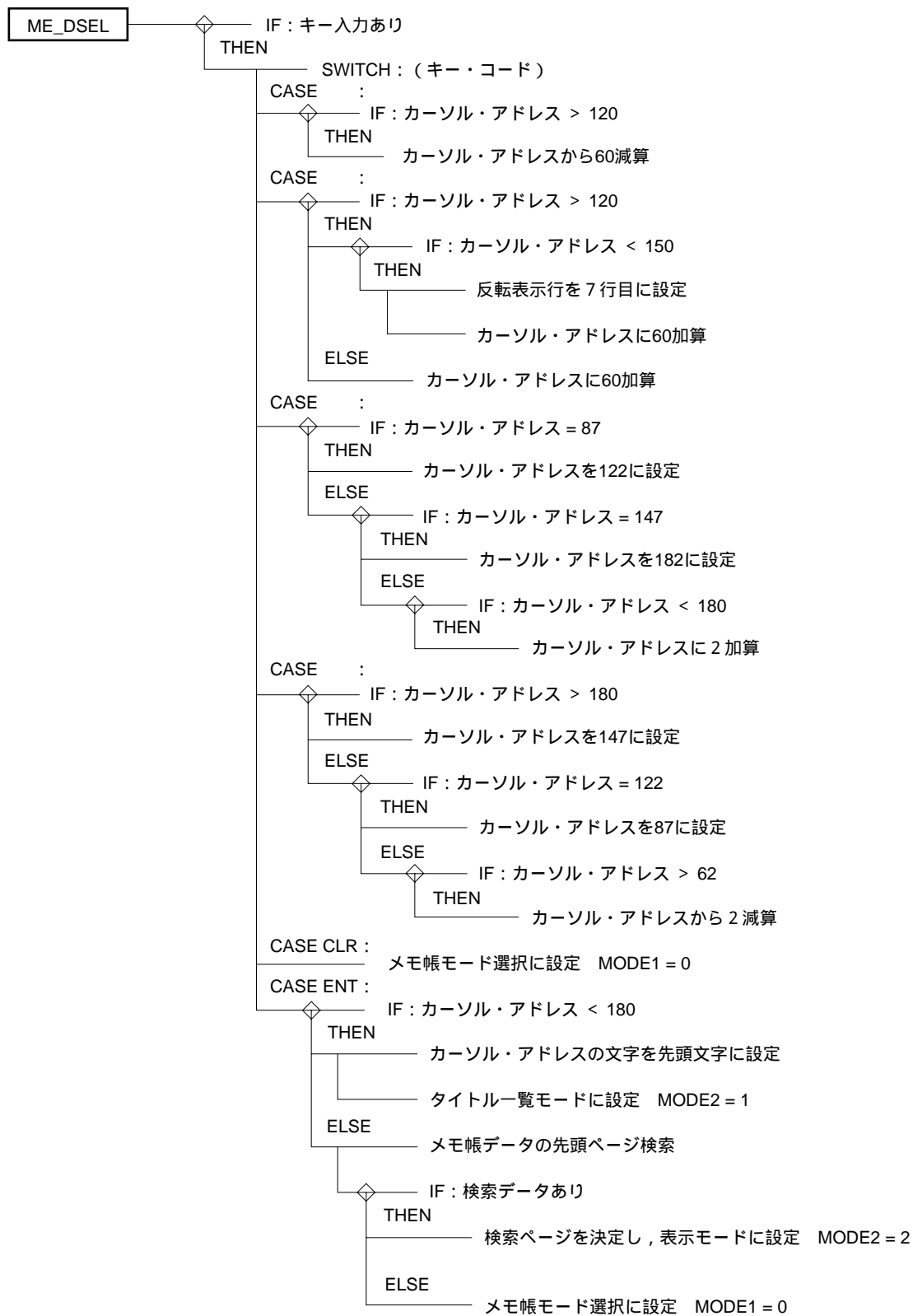
メモ帳モード

・削除モード



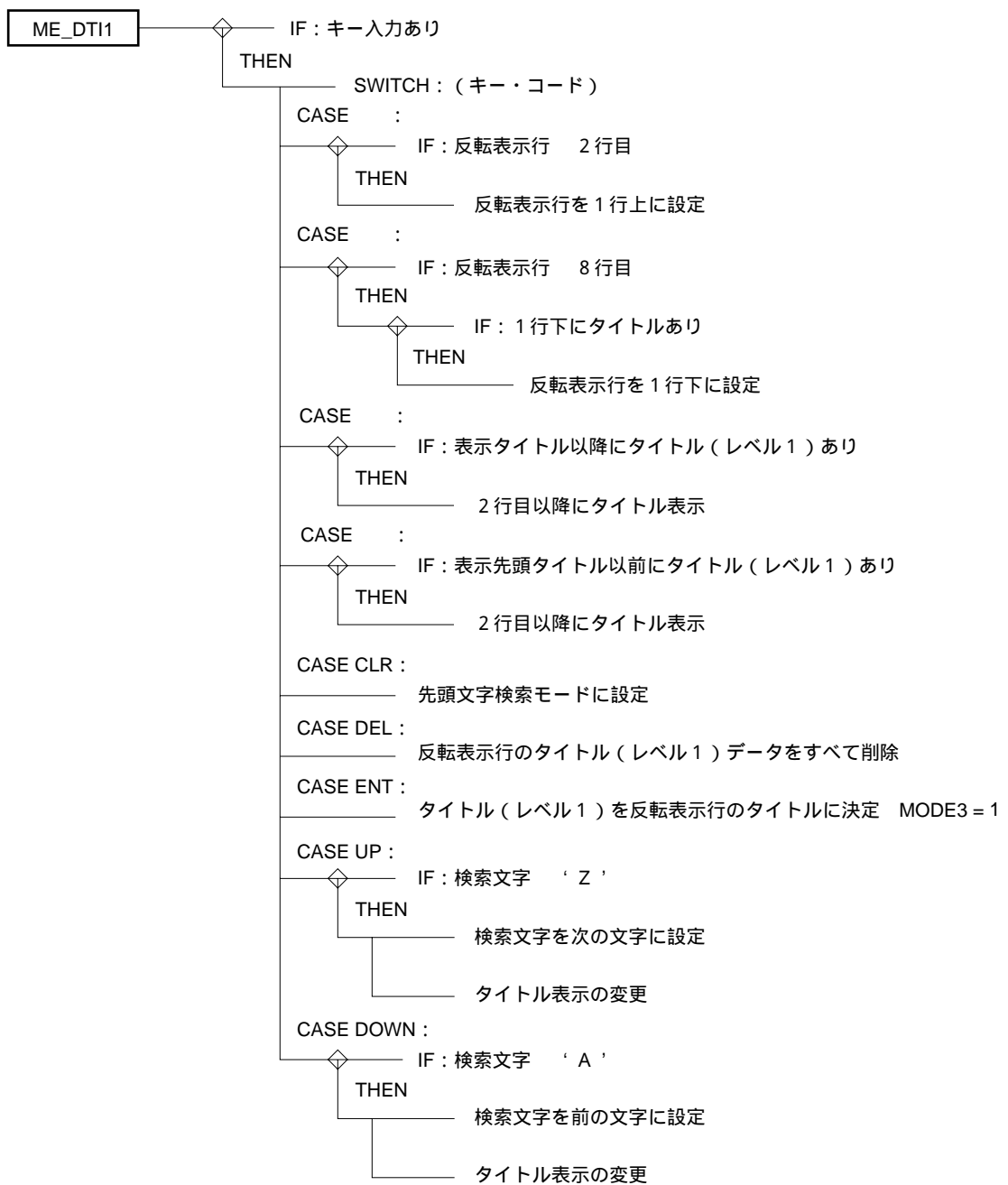
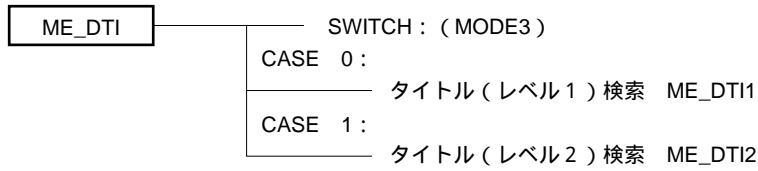
メモ帳モード

・削除モード



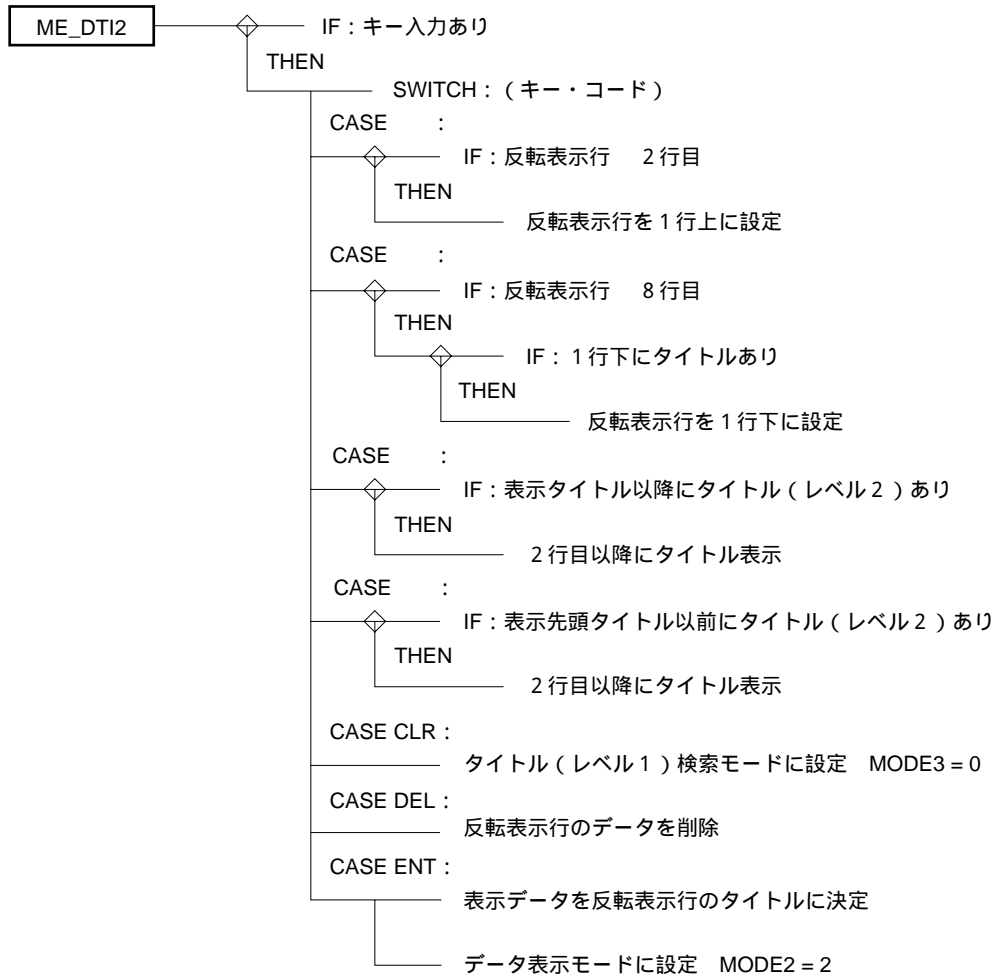
メモ帳モード

・削除モード



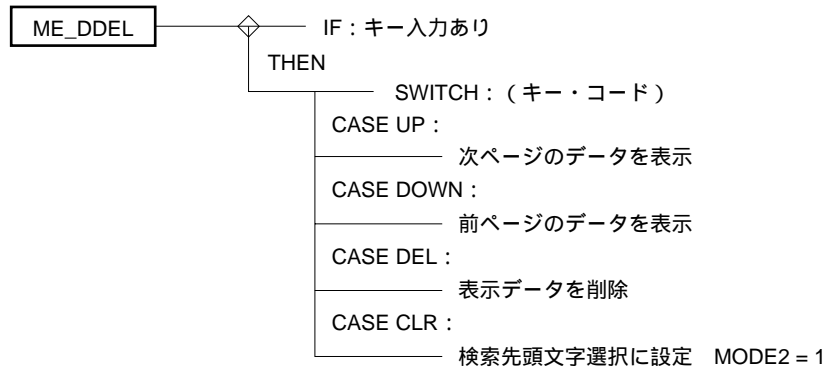
メモ帳モード

・削除モード

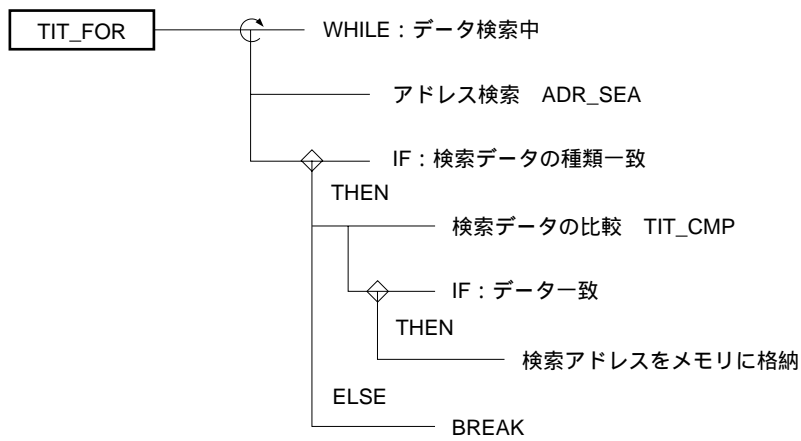


メモ帳モード

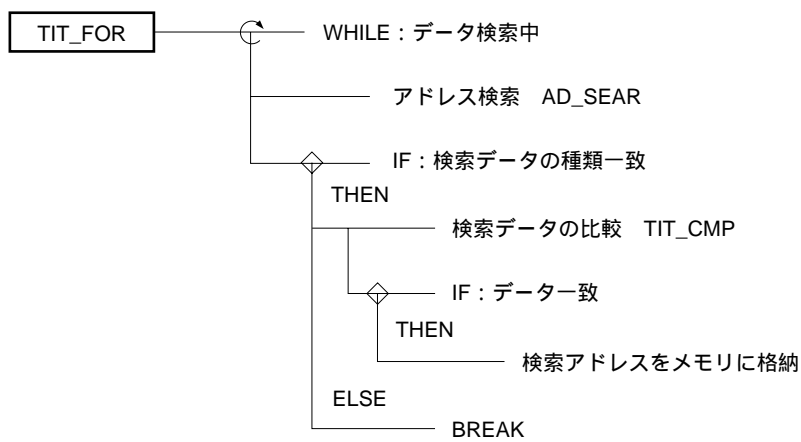
・削除モード



・タイトル検索 (前進)

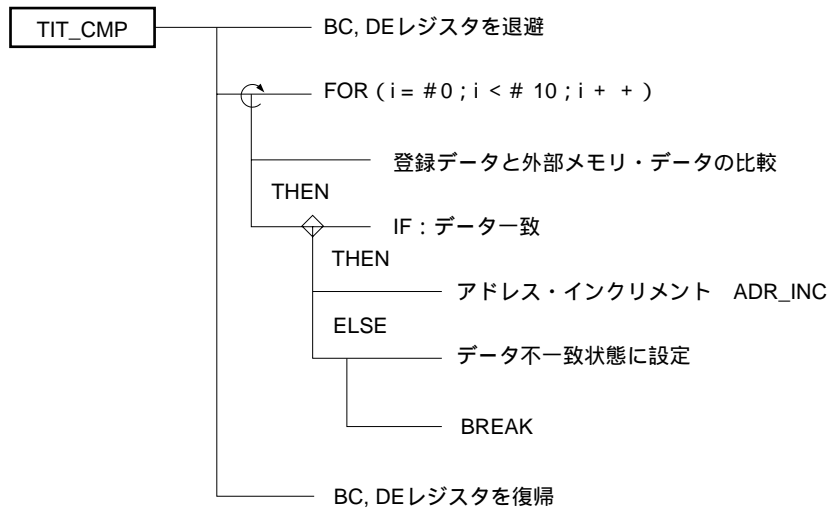


・タイトル検索 (後退)



メモ帳モード

・タイトル(レベル1)比較



4.7.9 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DM, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
EXTRN  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, MENPAG, MENBANK, BACPAG, BACBANK
EXTRN  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
EXTRN  COM_VOL, TRN_ADR, TRN_BNK, VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT, TIT_DAT
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DTILSET, DTIPSET, LCD_C, LCD_D, DT_SAV, STA_SEA, S_COMM
EXTRN  SAVADR, DTCOMP, DT_DEL, DEL_IL, BANKSET, ADR_SEA, AD_SEAR, ADR_INC, ADR_DEC, DT_SET
EXTRN  SEA_FOR, SEA_REV, SEA_FIL, NEX_ADR, BEF_ADR
EXTRN  ADRSTA, ADREND
EXTRN  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, IV_CHGF, COMM_F, DS_ERAF
EXTRN  CR_DSPF, ADENDF, ADSEAF, CMPERF, TR_CG1F, TR_CG2F, TR_CG3F, LCDCHGF

PUBLIC  S_MENO, MEM_VOL

$      INCLUDE (LCDCMD. INC)
$      INCLUDE (LCDT_EX. INC)

$LIST

          DSEG      SADDR
MEM_VOL:DS      1          ;メモ帳データ数
TIT_VOL:DS      1          ;メモ帳タイトル1 データ数
TIT_PNT:DS      1          ;タイトル(レベル1) 検索行データ
FOR_PNT:DS      1          ;タイトル(レベル1) 検索行(前通) 格納
REV_PNT:DS      1          ;タイトル(レベル1) 検索行(後通) 格納

          DSEG      IHRAM
TITLE1: DS      30         ;タイトル(レベル1) を格納
TITLE2: DS      30         ;タイトル(レベル2) を格納

;*****
;      メモ帳分岐処理
;*****
MENSEG      CSEG
S_MENO:
    switch(MODE1)
    case 0:
        CALL  !ME_SEL          ;モード選択
        break
    case 1:
        CALL  !ME_REG          ;データ登録
        break
    case 2:
        CALL  !ME_COR          ;データ修正
        break
    case 3:
        CALL  !ME_SEA          ;データ検索
        break
    case 4:
        CALL  !ME_DEL          ;データ削除
    ends
    RET

;*****
;      メモ帳モード選択処理
;*****
ME_SEL:
    if_bit(MD_CG1F)
        CLR1  MD_CG1F          ;メニュー表示データの設定
        HL=#LCDDS              ;LCD 1行目のデータを設定
        DE=#NEMODSP
        CALL  !DTILSET

        DE=#REG_D              ;LCD 2行目のデータを設定
        CALL  !DTILSET

        DE=#COR_D              ;LCD 3行目のデータを設定
        CALL  !DTILSET

```

```

DE=#SEA_D ;LCD 4行目のデータを設定
CALL !DTILSET

DE=#DEL_D ;LCD 5行目のデータを設定
CALL !DTILSET

DE=#CLR1LD ;LCD 6行目のデータを設定
CALL !DTILSET

DE=#CLR1LD ;LCD 7行目のデータを設定
CALL !DTILSET

DE=#CLR1LD ;LCD 8行目のデータを設定
CALL !DTILSET

MODE2=#0
MODE3=#0
MODE4=#0
SET1 MD_CG2F
SET1 MD_CG3F
SET1 MD_CG4F
CLR1 CR_DSPF ;反転表示行を2行目に設定
SET1 IV_CHGF
INV_DS=#2
SET1 LCDCHGF
endif

if_bit(COMM_F) ;通信状態
CALL !S_COMM ;通信処理
if_bit(!COMM_F) ;通信終了
SET1 MD_CG1F ;通信前の状態に設定
endif
else
if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD) ;キー・コード分岐
case CUR_UP:
if(INV_DS>#2)
INV_DS--
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CUR_DN: ;反転表示行を下に移動
if(INV_DS<#5)
INV_DS++
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CLEAR: ;前面面モードに変更
MODE0=#0
SET1 MD_CG0F
break
case ENTER: ;反転表示行のモード実行
A=INV_DS
A--
MODE1=A
SET1 MD_CG1F ;モード変更状態に設定
break
case COMM: ;通信状態に設定
COM_VOL=MEM_VOL (A) ;送信データ数の設定
B=#MEM_DAT ;検索データの設定
CALL !STA_SEA
TRN_ADR=H (A) ;送信データ数の設定
TRN_BNK=M_BANK (A)
SET1 COMM_F
SET1 TR_CG1F
ends
endif
endif
RET

;*****
; モモ帳データ登録分岐処理
;*****

```

```

ME_REG:
switch(MODE2)
case 0:
    if_bit(MD_CG2F)
        CLR1 MD_CG2F
        INV_DS=#0 ;反転表示削除
        SET1 IV_CHGF
        CALL !LCD_DSC ;LCD表示エリアをクリア
        HL=#LCDDS
        DE=#TIT_L1
        CALL !DTILSET ;0000H(1行目)に設定
        CUR_ADR=#30 ;カーソル・アドレスの設定
        SET1 CR_DSPF ;カーソル表示あり
        SET1 LCDCHGF
    endif
    CALL !ME_RKY0
    break
case 1:
    if_bit(MD_CG2F)
        CLR1 MD_CG2F
        HL=#LCDDS+60
        DE=#REG_L2
        CALL !DTILSET ;0000H+60(3行目)に設定
        CUR_ADR=#90 ;カーソル・アドレスの設定
        SET1 CR_DSPF ;カーソル表示あり
        SET1 LCDCHGF
    endif
    CALL !ME_RKY1
    break
case 2:
    if_bit(MD_CG2F)
        CLR1 MD_CG2F
        HL=#TITLE1 ;タイトル(レベル1)を登録
        DE=#LCDDS+30
        CALL !DTILSET

        HL=#TITLE2 ;タイトル(レベル2)を登録
        DE=#LCDDS+90
        CALL !DTILSET

        CALL !LCD_DSC
        HL=#LCDDS ;カーソル・アドレスの設定
        DE=#TITLE2 ;タイトル(レベル2)を表示
        CALL !DTILSET ;0000H(1行目)に設定

        HL=#LCDDS+30 ;0000H+30(2行目)に設定
        DE=#MEMO_D
        CALL !DTILSET
        SET1 DS_ERAF
        CLR1 CR_DSPF ;カーソル表示なし
        SET1 LCDCHGF
    endif
    CALL !ME_RKY2
    break
case 3:
    if_bit(MD_CG2F)
        CLR1 MD_CG2F
        CALL !LCD_DSC ;LCD表示エリアをクリア
        HL=#LCDDS+90 ;カーソル・アドレスの設定
        DE=#MEMO_OV ;エラー表示
        CALL !DTILSET ;0000H+90(4行目)に設定
        CLR1 CR_DSPF
        SET1 LCDCHGF
    endif
    CALL !ME_RKY3
ends
RET

;*****
; メモ帳データ登録キー入力処理
;*****
ME_RKY0:
    IF_BIT(KY_SIGF) ;キー入力あり
        CLR1 KY_SIGF
        IF(KEY_CD<#SPC || KEY_CD>#z') ;文字コード以外?

```

```

switch(KEY_CD)
case DELETE:
    B=CUR_ADR (A)           ;現在のカーソル・アドレスを設定
    C=CUR_ADR (A)           ;データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(C<#60) (A)
        [HL+B]=[HL+C] (A)   ;データ転送
        B++
        C++
    endw
    [HL+B]=#SPC (A)         ;行最終アドレスにスペース・データを設定
    break
case CLEAR:                 ;前モードに設定
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:                 ;タイトル(レベル2)入力モードに設定
    MODE2=#1
    SET1 MD_CG2F
    break
case S_DEL:                 ;1ライン・データ削除
    HL=#LCDDS+30           ;カーソル・アドレスの設定
    DE=#CLR1LD
    CALL !DTILSET
    CUR_ADR=#30           ;カーソル・アドレスを先頭に設定
    break
case CUR_RI:
    if(CUR_ADR<#59)        ;カーソル・アドレス2行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#30)        ;カーソル・アドレス2行目先頭アドレス以上
        CUR_ADR--
    endif
ends
ELSE
    B=CUR_ADR (A)           ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)       ;カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#59)        ;カーソル・アドレス2行目最終行以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

```

```

ME_RKY1:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
IF(KEY_CD<#SPC || KEY_CD<#'z') ;文字コード以外?
switch(KEY_CD)
case DELETE:
    B=CUR_ADR (A)           ;現在のカーソル・アドレスを設定
    C=CUR_ADR (A)           ;データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(C<#120) (A)
        [HL+B]=[HL+C] (A)   ;データ転送
        B++
        C++
    endw
    [HL+B]=#SPC (A)         ;行最終アドレスにスペース・データを設定
    break
case CLEAR:                 ;前モードに設定
    MODE1=#0
    SET1 MD_CG1F
    break
case ENTER:                 ;メモ入力モードに設定
    MODE2=#2
    SET1 MD_CG2F
    break
case S_DEL:                 ;1ライン・データ削除
    HL=#LCDDS+90           ;カーソル・アドレスの設定

```

```

        DE=#CLR1LD
        CALL !DTILSET
        CUR_ADR=#90 :カーソル・アドレスを先順に設定
        break
    case CUR_RI:
        if(CUR_ADR<#119) :カーソル・アドレス4行目最終行以内
            CUR_ADR++
        endif
        break
    case CUR_LE:
        if(CUR_ADR>#90) :カーソル・アドレス4行目先順アドレス以上
            CUR_ADR--
        endif
    ends
ELSE
    B=CUR_ADR (A) :カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A) :カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#119) :カーソル・アドレス4行目最終行以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET

NE_RKY2:
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
if_bit(DS_ERAF)
CLR1 DS_ERAF
HL=#LCDDS+30 :カーソル・アドレスの設定
DE=#CLR1LD
CALL !DTILSET :0 0 0 0 H + 6 0 (3行目)に設定
CUR_ADR=#30
SET1 CR_DSPF :カーソル表示あり
endif
IF(KEY_CD<#SPC || KEY_CD>#'z') :文字コード以外?
switch(KEY_CD)
case DELETE:
    B=CUR_ADR (A) :現在のカーソル・アドレスを設定
    C=CUR_ADR (A) :データ転送対象のアドレス(+1)を設定
    C++
    HL=#LCDDS
    while(C<#240) (A)
        [HL+B]=[HL+C] (A) :データ転送
        B++
        C++
    endw
    [HL+B]=#SPC (A) :行最終アドレスにスペース・データを設定
    break
case CLEAR: :消モードに設定
    MODE1=#0
    SET1 MD_CGIF
    break
CASE ENTER: :メモ・データの登録
    N=#ADRSTA
    L=#00H
    M_BANK=#0
    CALL !BANKSET
    B=#TIT_DAT :タイトル(レベル1)データの検索
    CLR1 ADENDF
    while_bit(!ADENDF) :検索アドレス決定
        CALL !ADR_SEA
        if_bit(!CY)
            CALL !BANKSET :バンク変更
        endif
    endw
    MENPAG=N (A) :検索アドレスの登録
    MENBNK=M_BANK (A) :検索メモリ・バンクの登録
    if([HL]==B) (A) :タイトル(レベル1)データあり
        CLR1 CNPBRF
        if(TIT_VOL1=#0)
            L=#10H
            j=TIT_VOL (A)
            while(j>#0)

```

```

C=#0
DE=#TITLE1
CALL !TIT_CMP
if_bit(CMPERF)
  if(L=#14) (A)
    CALL !ADR_INC
    if_bit(ICY)
      if(M_BANK<#4)
        CALL !BANKSET
      else
        MODE2=#3
        SET1 MD_CG2F
      endif
    endif
  else
    L=#00H
  endif
else
  L+=#10H (A)
endif
j--
else
  break
endif
endw
if_bit(CMPERF)
  CALL !TIT_SAV
endif
else
  CALL !TIT_SAV
endif
endif
CALL !TIT_SAV
endif
M_BANK=#3
CALL !BANKSET
A=!0BF00H
if(A=#VAC_DAT) ;最終アドレスのメモリ空きあり
  CLRI ADSEAF
  M_BANK=#0 ;外部メモリを開始位置に設定
  CALL !BANKSET
  H=#ADRSTA
  L=#0
  B=#MEM_DAT ;比較データをメモ帳データに設定
  C=#16 ;1ライン比較の開始アドレスを設定 (HL+C)
  DE=#LCDDS ;比較先頭アドレス
  CALL !SAVADR ;登録メモリ・アドレスを設定
  DE=#DTWORK ;転送用データの作成
  [DE]=#MEM_DAT (A) ;データ種類の設定
  DE++
  HL=#TITLE1 ;タイトル(レベル1)の設定
  for(i=#1;i<#11;i++) ;1-10バイトのデータをクリア
    [DE]=[HL] (A)
    DE++
    HL++
  next
  A=#0
  for(i=#11;i<#16;i++) ;11-15バイトのデータをクリア
    [DE]=A
    DE++
  next
  HL=#LCDDS ;LCD表示データをワーク・エリアに設定
  for(i=#0;i<#240;i++)
    [DE]=[HL] (A)
    DE++
    HL++
  next
  DE=#DTWORK
  H=#MEMPAG (A)
  L=#0
  CALL !DT_SAV ;データ登録
  MEM_VOL++
  MODE1=#0 ;登録終了
  SET1 MD_CG1F
else
  SET1 MD_CG2F ;メモリ・アドレスに空きなし

```

```

        MODE2=#3           ;オーバフロー・モードに設定
    endif
    break
case S_DEL:                ;1ライン・データ削除
    A=CUR_ADR              ;カーソル・アドレスの取り込み
    X=#0H
    A<->X
    C=#30                  ;1行文字数
    AX/=C                  ;行数算出(X)
    A=#30
    AX*=X                  ;先行アドレスの設定
    AX+=#LCDDS             ;カーソル・アドレスの設定
    HL=AX
    DE=#CLR1LD
    CALL !DT1LSET
    CUR_ADR=X(A)          ;カーソル・アドレスを先頭に設定
    break
case CUR_RI:
    if(CUR_ADR<#239)      ;カーソル・アドレス3行目最終行以内
        CUR_ADR++
    endif
    break
case CUR_LE:
    if(CUR_ADR>#30)       ;カーソル・アドレス3行目先頭アドレス以上
        CUR_ADR--
    endif
    break
case CUR_UP:
    if(CUR_ADR)=#60)      ;カーソル・アドレスを1行上(+30)に設定
        CUR_ADR-=#30
    endif
    break
case CUR_DN:
    if(CUR_ADR<#210)     ;カーソル・アドレスを1行下(-30)に設定
        CUR_ADR+=#30
    endif
endif
ends
ELSE
    B=CUR_ADR(A)          ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD(A)      ;カーソル・アドレス位置に文字データを設定
    if(CUR_ADR<#239)      ;カーソル・アドレス3行目最終行以内
        CUR_ADR++
    endif
endif
ENDIF
ENDIF
RET

ME_RKY3:
    if_bit(KY_SIGF)
    CLR1 KY_SIGF
    switch(KEY_CD)
    case CLEAR:           ;前モードに設定
        MODE1=#0
        SET1 MD_CG1F
    ends
endif
RET

;*****
;      メモ修正モード
;*****
ME_COR:
    if_bit(MD_CG1F)
    CLR1 MD_CG1F
    INV_DS=#0             ;反転表示削除
    SET1 IV_CHGF
    HL=#LCDDS             ;メニュー表示データの設定
    DE=#TIT_SEA
    CALL !DT1LSET        ;LCD1行目のデータを設定
    DE=#CLR1LD
    CALL !DT1LSET        ;LCD2行目のデータを設定
    DE=#ATOM_D

```

```

CALL    !DTILSET                ;LCD 3行目のデータを設定

DE=#CLR1LD
CALL    !DTILSET                ;LCD 4行目のデータを設定

DE=#NTOZ_D
CALL    !DTILSET                ;LCD 5行目のデータを設定

DE=#CLR1LD
CALL    !DTILSET                ;LCD 6行目のデータを設定

DE=#NEXT_P
CALL    !DTILSET                ;LCD 7行目のデータを設定

DE=#CLR1LD
CALL    !DTILSET                ;LCD 8行目のデータを設定

MODE2=#0
MODE3=#0
MODE4=#0
SET1    LCDCHGF

SET1    CR_DSPF                ;カーソル表示の設定
CUR_ADR=#62                    ;カーソル・アドレスを'A'の位置に設定
MENPAG=#ADRSTA
MEMBNK=#0
endif
switch(MODE2)
case 0:
CALL    !ME_CSEL                ;先頭文字選択モード
break
case 1:
CALL    !ME_CTI                 ;名前選択モード
break
case 2:
CALL    !ME_CCOR                ;データ修正モード
ends
RET

;*****
;      メモ帳先頭文字選択モード
;*****
ME_CSEL:
ME_SSEL:
ME_DSEL:
IF_BIT(KY_SIGF)                ;キー入力あり
CLR1    KY_SIGF
switch(KEY_CD)
case CUR_RI:
if(CUR_ADR==#86)                ;カーソル・アドレス=M
CUR_ADR=#122                    ;カーソル・アドレス=N
elseif(CUR_ADR==#146)          ;カーソル・アドレス=Z
CUR_ADR=#182                    ;7行目反転表示
INV_DS=#7
SET1    IV_CHGF
elseif(CUR_ADR<#180)           ;カーソル・アドレス上記以外の値
CUR_ADR+=#2                    ;カーソル・アドレスを次の文字に設定
endif
break
case CUR_LE:
if(CUR_ADR>#180)                ;反転表示行
CUR_ADR=#146                    ;カーソル・アドレス=Z
INV_DS=#0
SET1    IV_CHGF                ;反転表示消去
elseif(CUR_ADR==#122)          ;カーソル・アドレス=N
CUR_ADR=#86                    ;カーソル・アドレス=N
elseif(CUR_ADR>#62)            ;カーソル・アドレス上記以外の値
CUR_ADR=#2                    ;カーソル・アドレスを前の文字に設定
endif
break
case CUR_UP:
if(CUR_ADR>#120)                ;カーソル・アドレスN以降
CUR_ADR-=#60                    ;カーソル・アドレスを1行上(-60)に設定
INV_DS=#0
SET1    IV_CHGF                ;反転表示消去

```



```

endif
break
case CUR_DN:
  if(CUR_ADR<#120)           ;カーソル・アドレスN以降
    if(CUR_ADR<#150)       ;反転表示でない
      INV_DS=#7
      SETI  IV_CHGF        ;7行目反転表示
      CUR_ADR+=#60        ;カーソル・アドレスを1行下(+60)に設定
    endif
  else
    CUR_ADR+=#60           ;カーソル・アドレスを1行下(+60)に設定
  endif
  break
case CLEAR:                 ;前モードに設定
  MODE1=#0
  SETI  ND_CG1F
  break
case ENTER:
  if(CUR_ADR<#180)         ;カーソル・アドレス=A-Z
    HL=#LCDDS
    B=CUR_ADR (A)
    SEACHR=[HL+B] (A)      ;検索文字を登録
    SETI  ND_CG2F
    SETI  ND_CG3F
    SETI  ND_CG4F
    SETI  LCDCHGF
    NODE2=#1              ;タイトル選択モードに設定
    NODE3=#0              ;タイトル(レベル1)選択モードに設定
    NODE4=#0
    BACPAG=MEMPAG (A)
    BACBNK=MEMBNK (A)
    CALL  !LCD_DSC
  else
    B=#MEM_DAT            ;メモ帳のデータ検出設定
    CALL  !STA_SEA        ;先頭ページの検出
    if([HL]==#MEM_DAT) (A) ;検索アドレスのメモ帳データ
      MEMBNK=M_BANK (A) ;メモリ・アドレスを登録
      MEMPAG=M (A)
      NODE2=#2            ;データ修正モードに設定
      SETI  ND_CG2F
    else
      NODE1=#0            ;メモ帳モード選択に設定
      SETI  ND_CG1F
    endif
  endif
endif
ends
ENDIF
RET

;*****
;   メモ帳タイトル選択モード
;*****
ME_CTI:
ME_STI:
  switch(NODE3)            ;検索タイトルの指定
  case 0:
    CALL  !ME_CTI1        ;タイトル(レベル1)一覧
    break
  case 1:
    CALL  !ME_CTI2        ;タイトル(レベル2)一覧
  ends
  RET

;*****
;   メモ帳タイトル(レベル1)選択モード
;*****
ME_CTI1:
  if_bit(ND_CG3F)
  CLRI  ND_CG3F
  CLRI  CR_DSPF
  CALL  !TI1_DSP
  FOR_BNK=#0
  REV_BNK=#0
  FOR_ADR=#ADRSTA
  REV_ADR=#ADRSTA

```

```

FOR_PNT=#10H
MODE4=#0
SET1 MD_CG4F
endif
IF_BIT(MD_CG4F)
CLR1 MD_CG4F
SWTCH(MODE4)
CASE 0:
  FOR(j=#0;j<#7;j++)          ;1 ページ表示数のカウント
    H=FOR_ADR (A)
    L=#00H
    M_BANK=FOR_BNK (A)        ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    TIT_PNT=FOR_PNT (A)
    B=#TIT_DAT
    CALL !TIT_FAD
    if_bit(ADSEAF)
      D=H (A)                  ;転送元アドレスの設定
      E=TIT_PNT (A)
      PUSH DE                  ;アドレス退避
      swtch(j)                 ;転送先アドレスの設定
      case 0:                   ;2 行目に設定
        CALL !TIT_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
        REV_PNT=TIT_PNT (A)
        CALL !TIT_INC
        CALL !TIT_DSP
        HL=#LCDDS+80
        break
      case 1:                   ;3 行目に設定
        HL=#LCDDS+60
        break
      case 2:                   ;4 行目に設定
        HL=#LCDDS+90
        break
      case 3:                   ;5 行目に設定
        HL=#LCDDS+120
        break
      case 4:                   ;6 行目に設定
        HL=#LCDDS+150
        break
      case 5:                   ;7 行目に設定
        HL=#LCDDS+180
        break
      case 6:                   ;8 行目に設定
        HL=#LCDDS+210
      ends
      B=#10
      POP DE
      CALL !DT_SET
      H=D (A)
      CALL !TIT_INC
      FOR_ADR=H (A)
      FOR_BNK=M_BANK (A)
      FOR_PNT=TIT_PNT (A)
    else                         ;検索データなし
      break
    endif
  NEXT
  break
CASE 1:
  FOR(j=#0;j<#7;j++)          ;1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)        ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    TIT_PNT=REV_PNT (A)
    B=#TIT_DAT
    CALL !TIT_RAD
    if_bit(ADSEAF)
      D=H (A)                  ;転送元アドレスの設定
      E=TIT_PNT (A)
      PUSH DE                  ;アドレス退避
      swtch(j)                 ;転送先アドレスの設定

```

```

case 0: ; 8行目に設定
CALL ITIT_INC
FOR_ADR=H (A)
FOR_BNK=M_BANK (A)
FOR_PNT=TIT_PNT (A)
CALL ITIT_DEC
CALL ITI1_DSP
HL=#LCDDS+210
break
case 1: ; 7行目に設定
HL=#LCDDS+180
break
case 2: ; 6行目に設定
HL=#LCDDS+150
break
case 3: ; 5行目に設定
HL=#LCDDS+120
break
case 4: ; 4行目に設定
HL=#LCDDS+90
break
case 5: ; 3行目に設定
HL=#LCDDS+60
break
case 6: ; 2行目に設定
HL=#LCDDS+30
ends
B=#10
POP DE
CALL !DT_SET
H=D (A)
CALL ITIT_DEC
REV_ADR=H (A)
REV_BNK=M_BANK (A)
REV_PNT=TIT_PNT (A)
elseif_bit(ADENDF) ; 検索データなし
break
endif
NEXT
ENDS
ENDIF

IF_BIT(KY_SIGF) ; キー入力あり
CLR1 KY_SIGF
switch(KEY_CD)
case CUR_UP: ; 反転表示行を上に移動
if(INV_DS>#2)
INV_DS--
SET1 IV_CHGF ; 反転表示行変更状態に設定
endif
break
case CUR_DN: ; 反転表示行を下に移動
if(INV_DS<#8)
switch(INV_DS) ; 転送先アドレスの設定
case 2: ; 3行目に設定
HL=#LCDDS+60
break
case 3: ; 4行目に設定
HL=#LCDDS+90
break
case 4: ; 5行目に設定
HL=#LCDDS+120
break
case 5: ; 6行目に設定
HL=#LCDDS+150
break
case 6: ; 7行目に設定
HL=#LCDDS+180
break
case 7: ; 8行目に設定
HL=#LCDDS+210
ends
if([HL]!="#SPC) (A) ; 次の行に名前あり
INV_DS++
SET1 IV_CHGF ; 反転表示行変更状態に設定

```

```

        endif
    endif
    break
case CUR_RI:
    MODE4=#1
    SET1 MD_CG4F
    break
case CUR_LE:
    MODE4=#0
    SET1 MD_CG4F
    break
case CLEAR:
    MODE1=#2
    SET1 MD_CG1F
    break
case ENTER:
    switch(INV_DS) ;検索対象行アドレスの設定
    case 2: ;2行目に設定
        DE=#LCDDS+30
        break
    case 3: ;3行目に設定
        DE=#LCDDS+60
        break
    case 4: ;4行目に設定
        DE=#LCDDS+90
        break
    case 5: ;5行目に設定
        DE=#LCDDS+120
        break
    case 6: ;6行目に設定
        DE=#LCDDS+150
        break
    case 7: ;7行目に設定
        DE=#LCDDS+180
        break
    case 8: ;8行目に設定
        DE=#LCDDS+210
    ends
    if([DE]=#SPC) (A)
        MODE3=#1
        MODE4=#0
        SET1 MD_CG3F
        SET1 MD_CG4F
        HL=#TITLE1
        B=#10
        CALL !DT_SET
        CALL !LCD_DSC
    ;
    else
    ;
        SET1 MD_CG1F

    endif
    break
case UP:
    if(SEACHR<#'Z') ;検索文字'A'-'Y'
        SET1 MD_CG3F
        SEACHR++ ;次の文字に変更
    endif
    break
case DOWN:
    if(SEACHR>#'A') ;検索文字'B'-'Z'
        SET1 MD_CG3F
        SEACHR-- ;前の文字に変更
    endif
ends
ENDIF
RET

;*****
;メモ帳タイトル(レベル2)選択モード
;*****
ME_CT12:
    if_bit(MD_CG3F)

```

```

CLR1 MD_CG3F
CALL !TI2_DSP
FOR_BNK=#0
FOR_ADR=#ADRSTA
REV_BNK=#0
REV_ADR=#ADRSTA
MODE4=#0
SET1 MD_CG4F
endif
IF_BIT(MD_CG4F)
CLR1 MD_CG4F
SWITCH(MODE4)
CASE 0:
  FOR(j=#0;j<#7;j++) :1 ページ表示数のカウント
    H=FOR_ADR (A)
    L=#00H
    M_BANK=FOR_BNK (A) :現在のメモリ・アドレスを設定
    CALL !BANKSET
    B=#MEM_DAT
    CALL !TIT_FOR
    if_bit(ADSEAF)
      D=H (A) :転送元アドレスの設定
      E=#16
      PUSH DE :アドレス退避
      switch(j) :転送先アドレスの設定
      case 0: :2 行目に設定
        CALL !NEX_ADR
        CALL !TI2_DSP
        HL=#LCDDS+30
        break
      case 1: :3 行目に設定
        HL=#LCDDS+60
        break
      case 2: :4 行目に設定
        HL=#LCDDS+90
        break
      case 3: :5 行目に設定
        HL=#LCDDS+120
        break
      case 4: :6 行目に設定
        HL=#LCDDS+150
        break
      case 5: :7 行目に設定
        HL=#LCDDS+180
        break
      case 6: :8 行目に設定
        HL=#LCDDS+210
      ends
      POP DE
      CALL !DT1LSET
      H=D (A)
      CALL !ADR_INC
      FOR_ADR=H (A)
      FOR_BNK=M_BANK (A)
    elseif_bit(ADENDF) :検索データなし
      break
    endif
  NEXT
  break
CASE 1:
  FOR(j=#0;j<#7;j++) :1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A) :現在のメモリ・アドレスを設定
    CALL !BANKSET
    B=#MEM_DAT
    C=#1
    DE=#TITLE1
    CALL !TIT_REV
    if_bit(ADSEAF)
      D=H (A) :転送元アドレスの設定
      E=#16
      PUSH DE :アドレス退避
      switch(j) :転送先アドレスの設定
      case 0: :8 行目に設定

```

```

        CALL 1BEF_ADR
        CALL !TI2_DSP
        HL=#LCDDS+210
        break
    case 1: ; 7行目に設定
        HL=#LCDDS+180
        break
    case 2: ; 6行目に設定
        HL=#LCDDS+150
        break
    case 3: ; 5行目に設定
        HL=#LCDDS+120
        break
    case 4: ; 4行目に設定
        HL=#LCDDS+90
        break
    case 5: ; 3行目に設定
        HL=#LCDDS+60
        break
    case 6: ; 2行目に設定
        HL=#LCDDS+30
        INV_DS=#2
        SET1 IV_CHGF
    ends
    POP DE
    CALL !DTILSET
    H=D(A)
    CALL !ADR_DEC
    REV_ADR=H(A)
    REV_BNK=M_BANK(A)
elseif_bit(ADENDF) ; 検索データなし
    break
endif
NEXT
ENDS
ENDIF

IF_BIT(KY_SIGF) ; キー入力あり
CLR1 KY_SIGF
switch(KEY_CD)
case CUR_UP: ; 反転表示行を上に移動
    if(INV_DS>#2)
        INV_DS--
        SET1 IV_CHGF ; 反転表示行変更状態に設定
    endif
    break
case CUR_DN: ; 反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS) ; 転送先アドレスの設定
        case 2: ; 3行目に設定
            HL=#LCDDS+60
            break
        case 3: ; 4行目に設定
            HL=#LCDDS+90
            break
        case 4: ; 5行目に設定
            HL=#LCDDS+120
            break
        case 5: ; 6行目に設定
            HL=#LCDDS+150
            break
        case 6: ; 7行目に設定
            HL=#LCDDS+180
            break
        case 7: ; 8行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!=#SPC)(A) ; 次の行に名前あり
            INV_DS++
            SET1 IV_CHGF ; 反転表示行変更状態に設定
        endif
    endif
    break
case CUR_RI:
    MODE4=#1

```

```

        SET1 MD_CG4F
        break
case CUR_LE:
    MODE4=#0
    SET1 MD_CG4F
    break
case CLEAR:
    MODE1=#2
    SET1 MD_CG1F
    break
CASE ENTER:

```

```

H=#ADRSTA
L=#00H
M_BANK=#0 ;現在のメモリ・アドレスを設定
CALL !BANKSET
B=#MEM_DAT ;検索データの設定
while(forever)
    CALL !TIT_FOR ;タイトル(レベル1)の検索
    if_bit(ADSEAF) ;データあり
        switch(INV_DS) ;検索対象行アドレスの設定
            case 2: ;2行目に設定
                DE=#LCDDS+30
                break
            case 3: ;3行目に設定
                DE=#LCDDS+60
                break
            case 4: ;4行目に設定
                DE=#LCDDS+90
                break
            case 5: ;5行目に設定
                DE=#LCDDS+120
                break
            case 6: ;6行目に設定
                DE=#LCDDS+150
                break
            case 7: ;7行目に設定
                DE=#LCDDS+180
                break
            case 8: ;8行目に設定
                DE=#LCDDS+210
                break
        ends
        C=#16
        CALL !DTCOMP ;タイトル(レベル2)一致
        if_bit(ICMPERF)
            MEMBNK=M_BANK(A)
            MEMPAG=H(A)
            MODE2=#2 ;登録データあり→データ表示
            SET1 MD_CG2F
            break
        else
            CALL !ADR_INC
            if_bit(ICY)
                if(M_BANK<=#3)
                    CALL !BANKSET ;バンク変更
                else
                    break
                endif
            endif
        else
            break
        endif
    endwhile

;
; CLR1 ADSEAF
; M_BANK=#0 ;外部メモリを開始位置に設定
; CALL !BANKSET
; H=#ADRSTA
; L=#0
; B=#MEM_DAT ;比較データを名刺(個人)データに設定
; C=#16 ;1ライン比較の開始アドレスを設定(HL+C)
; switch(INV_DS) ;検索対象行アドレスの設定

```

```

;           case 2:           ; 2行目に設定
;           DE=#LCDDS+30
;           break
;           case 3:           ; 3行目に設定
;           DE=#LCDDS+60
;           break
;           case 4:           ; 4行目に設定
;           DE=#LCDDS+90
;           break
;           case 5:           ; 5行目に設定
;           DE=#LCDDS+120
;           break
;           case 6:           ; 6行目に設定
;           DE=#LCDDS+150
;           break
;           case 7:           ; 7行目に設定
;           DE=#LCDDS+180
;           break
;           case 8:           ; 8行目に設定
;           DE=#LCDDS+210
;           ends
;
;           if([DE]!="#SPC) (A)
;           repeat
;           CALL !SAVADR       ;メモリ・アドレスを設定
;           if_bit(CY)
;           CALL !ADR_INC
;           endif
;           if([HL]==#VAC_DAT) (A) ;検索アドレスのデータは空きデータ
;           break             ;登録データなし→検索終了
;           else
;           MODE2=#2          ;登録データあり→データ表示
;           SET1 MD_CG2F
;           endif
;           until_bit(ICHPERF)
;
;           MEMBNK=M_BANK (A)  ;
;           MEMPAG=M (A)
;           endif
;           ends
;           ENDIF
;           RET
;
;*****
;           メモ帳修正モード
;*****
;           MB_CCOR:
;           if_bit(MD_CG2F)    ;修正データの変更
;           CLR1 MD_CG2F
;           M_BANK=MEMBNK (A) ;メモリ・アドレスの設定
;           CALL !BANKSET
;           D=MEMPAG (A)
;           E=#16
;           HL=#LCDDS        ;LCD表示アドレスの設定
;           CALL !DT1PSET    ;表示データの転送
;           CUR_ADR=#30      ;カーソル・アドレスを1文字目に設定
;           SET1 CR_DSPF
;           INV_DS=#0        ;反転表示行削除
;           SET1 IV_CHGF
;           SET1 LCDCHGF
;           endif
;           IF_BIT(KY_SIGF)
;           CLR1 KY_SIGF
;           IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
;           switch(KEY_CD)
;           case DELETE:
;           B=CUR_ADR (A)    ;現在のカーソル・アドレスを設定
;           C=CUR_ADR (A)   ;データ転送対象のアドレス(+1)を設定
;           C++
;           HL=#LCDDS
;           while(C<#240) (A)
;           [HL+B]=[HL+C] (A) ;データ転送
;           B++
;           C++
;           endwhile
;           endw

```



```

[HL+B]=#SPC (A) ;行最終アドレスにスペース・データを設定
break
case CLEAR: ;前モードに設定
MODE2=#0
SET1 MD_CG1F
FOR_ADR=#ADRSTA
REV_ADR=#ADRSTA
FOR_BNK=#0
REV_BNK=#0
break
case ENTER: ;修正データの登録
M_BANK=MEMBNK (A)
CALL !BANKSET
H=MEMPAG (A)
L=#16
DE=#LCDDS
CALL !DT1PSET
MODE1=#0
SET1 MD_CG1F
break
case S_DEL: ;1ライン・データ削除
A=CUR_ADR ;カーソル・アドレスの取り込み
X=#0H
A<->X
C=#30 ;1行文字数
AX/=C ;行数算出(X)
A=#30
AX*=X ;行先頭アドレスの設定
AX+=#LCDDS ;カーソル・アドレスの設定
HL=AX
DE=#CLR1LD
CALL !DT1LSET
CUR_ADR=X (A) ;カーソル・アドレスを先頭に設定
break
case CUR_UP:
if(CUR_ADR<#239) ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
break
case CUR_DOWN:
if(CUR_ADR>#30) ;カーソル・アドレス2行目先頭アドレス以上
CUR_ADR--
endif
break
case CUR_LEFT:
if(CUR_ADR=#80) ;カーソル・アドレスを1行上(+30)に設定
CUR_ADR-=#30
endif
break
case CUR_RIGHT:
if(CUR_ADR<#210) ;カーソル・アドレスを1行下(-30)に設定
CUR_ADR+=#30
endif
break
case UP:
H=MEMPAG (A)
L=#00H
M_BANK=MEMBNK (A) ;現在のメモリ・アドレスを設定
CALL !BANKSET
if(H=#ADREND) (A) ;バンク最終アドレス
if(M_BANK<#3) ;バンク1-バンク3
M_BANK++ ;バンク更新
CALL !BANKSET
H=#ADRSTA
if([HL]==#MEM_DAT) (A) ;次データ=メモ帳データ
SET1 MD_CG2F
MEMPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H++ ;次データ参照
if([HL]==#MEM_DAT) (A)
SET1 MD_CG2F ;アドレスの更新
MEMPAG=H (A)
endif
endif

```

```

MEMBNK=M_BANK (A)
endif
endif
break
case DOWN:
H=MEMPAG (A)
L=#00H
M_BANK=MEMBNK (A) ;現在のメモリ・アドレスを設定
CALL !BANKSET
if(H==#ADRSTA) (A) ;バンク先頭アドレス
if(M_BANK>#0) ;バンク1-バンク9
M_BANK-- ;バンク更新
CALL !BANKSET
H=#ADREND
if([HL]==#MEM_DAT) (A) ;次データ=メモ帳データ
SETI MD_CG2F
MEMPAG=H (A) ;アドレスの更新
MEMBNK=M_BANK (A)
endif
endif
else
H-- ;次データ参照
if([HL]==#MEM_DAT) (A)
SETI MD_CG2F ;アドレスの更新
MEMPAG=H (A)
MEMBNK=M_BANK (A)
endif
endif
ends
ELSE
B=CUR_ADR (A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A) ;カーソル・アドレス位置に文字データを設定
if(CUR_ADR<#239) ;カーソル・アドレス1行目最終行以内
CUR_ADR++
endif
ENDIF
ENDIF
RET

;*****
; モモ帳先頭文字選択モード
;*****
ME_SEA:
if_bit(MD_CG1F)
CLR1 MD_CG1F
HL=#LCDDS ;メニュー表示データの設定
DE=#TIT_SEA
CALL !DTILSET ;LCD 1行目のデータを設定

DE=#CLR1LD
CALL !DTILSET ;LCD 2行目のデータを設定

DE=#ATOM_D
CALL !DTILSET ;LCD 3行目のデータを設定

DE=#CLR1LD
CALL !DTILSET ;LCD 4行目のデータを設定

DE=#NTOZ_D
CALL !DTILSET ;LCD 5行目のデータを設定

DE=#CLR1LD
CALL !DTILSET ;LCD 6行目のデータを設定

DE=#NEXT_P
CALL !DTILSET ;LCD 7行目のデータを設定

DE=#CLR1LD
CALL !DTILSET ;LCD 8行目のデータを設定

MODE2=#0
MODE3=#0
MODE4=#0

```

```

SET1 CR_DSPF ;カーソル表示の設定
CUR_ADR=#62 ;カーソル・アドレスを'A'の位置に設定
INV_DS=#0 ;反転表示行削除
SET1 IV_CHGF
SET1 LCDCHGF
endif
switch(MODE2)
case 0:
CALL IME_SSEL ;先頭文字選択モード
break
case 1:
CALL IME_STI ;タイトル選択モード
break
case 2:
CALL IME_SSEA ;データ表示モード
ends
RET

;*****
; メモ帳検索モード
;*****
ME_SSEA:
if_bit(COMM_F) ;通信状態
CALL IS_COMM ;通信処理
if_bit(!COMM_F) ;通信終了
SET1 MD_CG2F ;通信前の状態に設定
endif
else
if_bit(MD_CG2F) ;修正データの変更
CLR1 MD_CG2F
M_BANK=MEMBANK (A) ;メモリ・アドレスの設定
CALL !BANKSET
D=MEMPAG (A)
E=#16
HL=#LCDDS ;LCD表示アドレスの設定
CALL !DT1PSET ;表示データの転送
INV_DS=#0 ;反転表示行削除
SET1 IV_CHGF
CLR1 CR_DSPF
SET1 LCDCHGF
endif
IF_BIT(KY_SIGF)
CLR1 KY_SIGF
IF(KEY_CD<#SPC || KEY_CD>#'z') ;文字コード以外?
switch(KEY_CD)
case CLEAR: ;前モードに設定
MODE2=#0
SET1 MD_CG1F
FOR_ADR=#ADRSTA
REV_ADR=#ADRSTA
FOR_BNK=#0
REV_BNK=#0
break
case UP:
H=MEMPAG (A)
L=#00H
M_BANK=MEMBANK (A) ;現在のメモリ・アドレスを設定
CALL !BANKSET
if(H==#ADREND) (A) ;バンク最終アドレス
if(M_BANK<#3) ;バンク1-バンク3
M_BANK++ ;バンク更新
CALL !BANKSET
H=#ADRSTA
if([HL]==#MEM_DAT) (A) ;次データ=メモ帳データ
SET1 MD_CG2F
MEMPAG=H (A) ;アドレスの更新
MEMBANK=M_BANK (A)
endif
endif
else
H++ ;次データ参照
if([HL]==#MEM_DAT) (A)
SET1 MD_CG2F ;アドレスの更新
MEMPAG=H (A)
MEMBANK=M_BANK (A)
endif
endif
endif

```

```

        endif
    endif
    break
case DOWN:
    H=MEMPAG (A)
    L=#00H
    M_BANK=MEMBNK (A) ;現在のメモリ・アドレスを設定
    CALL !BANKSET
    if(H==#ADRSTA) (A) ;バンク先頭アドレス
        if(M_BANK>#0) ;バンク1-バンク3
            M_BANK-- ;バンク更新
            CALL !BANKSET
        H=#ADREND
        if([HL]==#MEM_DAT) (A) ;次データ=メモ帳データ
            SETI MD_CG2F
            MEMPAG=H (A) ;アドレスの更新
            MEMBNK=M_BANK (A)
        endif
    endif
else
    H-- ;次データ参照
    if([HL]==#MEM_DAT) (A)
        SETI MD_CG2F ;アドレスの更新
        MEMPAG=H (A)
        MEMBNK=M_BANK (A)
    endif
endif
break
case COMM:
    TRN_ADR=MEMPAG (A) ;データ送信アドレスの設定
    TRN_BNK=MEMBNK (A)
    COM_VOL=#1 ;データ送信数1バイトの設定
    SETI COMM_F
    SETI TR_CG1F
ends
endif
ENDIF
endif
RET

;*****
; タイトル先頭文字選択モード
;*****
ME_DEL:
    if_bit(MD_CG1F)
        CLR1 MD_CG1F
        HL=#LCDDS ;メニュー表示データの設定
        DE=#TIT_SEA
        CALL !DTILSET ;LCD 1行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET ;LCD 2行目のデータを設定

        DE=#ATOM_D
        CALL !DTILSET ;LCD 3行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET ;LCD 4行目のデータを設定

        DE=#NTOZ_D
        CALL !DTILSET ;LCD 5行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET ;LCD 6行目のデータを設定

        DE=#NEXT_P
        CALL !DTILSET ;LCD 7行目のデータを設定

        DE=#CLR1LD
        CALL !DTILSET ;LCD 8行目のデータを設定

        NODE2=#0
        NODE3=#0
        NODE4=#0
        SETI LCDCHGF

```

```

        SET1    CR_DSPF          ;カーソル表示の設定
        CUR_ADR=#62            ;カーソル・アドレスを'A'の位置に設定
        !NV_DS=#0             ;反転表示行削除
        SET1    IV_CHGF

    endif
    switch(MODE2)
    case 0:
        CALL    !ME_DSEL        ;先頭文字選択モード
        break
    case 1:
        CALL    !ME_DTI         ;名前選択モード
        break
    case 2:
        CALL    !ME_DDEL        ;データ表示モード
    ends
    RET

;*****
;   タイトル選択モード(削除)
;*****
ME_DTI:
    switch(MODE3)              ;検索タイトルの指定
    case 0:
        CALL    !ME_DTI1       ;タイトル(レベル1)一覧
        break
    case 1:
        CALL    !ME_DTI2       ;タイトル(レベル2)一覧
    ends
    RET

;*****
;   メモ帳タイトル(レベル1)削除モード
;*****
ME_DTI1:
    if_bit(MD_CG3F)
        CLR1    MD_CG3F
        CLR1    CR_DSPF

        CALL    !TII_DSP
        FOR_BNK=#0
        FOR_ADR=#ADRSTA
        REV_BNK=#0
        REV_ADR=#ADRSTA
        FOR_PNT=#10H
        MODE4=#0
        SET1    MD_CG4F
    endif
    IF_BIT(MD_CG4F)
        CLR1    MD_CG4F
        SWITCH(MODE4)
        CASE 0:
            FOR(j=#0;j<#7;j++)      ;1ページ表示数のカウント
                H=FOR_ADR(A)
                L=#00H
                M_BANK=FOR_BNK(A)   ;現在のメモリ・アドレスを設定
                CALL    !BANKSET
                TIT_PNT=FOR_PNT(A)
                B=#TIT_DAT
                CALL    !TIT_FAD
                if_bit(ADSEAF)
                    D=H(A)          ;転送元アドレスの設定
                    E=TIT_PNT(A)
                    PUSH    DE       ;アドレス過渡
                    switch(j)
                    case 0:
                        CALL    !TIT_DEC
                        REV_ADR=H(A)
                        REV_BNK=M_BANK(A)
                        REV_PNT=TIT_PNT(A)
                        CALL    !TIT_INC
                        CALL    !TII_DSP
                        HL=#LCDDS+30
                        break
                    case 1:
                        ;3行目に設定

```

```

        HL=#LCDDS+60
        break
    case 2:                                ; 4行目に設定
        HL=#LCDDS+90
        break
    case 3:                                ; 5行目に設定
        HL=#LCDDS+120
        break
    case 4:                                ; 6行目に設定
        HL=#LCDDS+150
        break
    case 5:                                ; 7行目に設定
        HL=#LCDDS+180
        break
    case 6:                                ; 8行目に設定
        HL=#LCDDS+210
    ends
    POP    DE
    B=#10
    CALL  !DT_SET
    H=D (A)
    CALL  !TIT_INC
    FOR_ADR=H (A)
    FOR_BNK=M_BANK (A)
    FOR_PNT=TIT_PNT (A)
else
    break                                ; 検索データなし
endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)                       ; 1ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)                    ; 現在のメモリ・アドレスを設定
    CALL  !BANKSET
    TIT_PNT=REV_PNT (A)
    B=#TIT_DAT
    CALL  !TIT_RAD
    if_bit(ADSEAF)
        D=H (A)                          ; 転送元アドレスの設定
        E=TIT_PNT (A)
        PUSH  DE                          ; アドレス過避
        switch(j)                         ; 転送先アドレスの設定
        case 0:
            CALL  !TIT_INC
            FOR_ADR=H (A)
            FOR_BNK=M_BANK (A)
            FOR_PNT=TIT_PNT (A)
            CALL  !TIT_DEC
            CALL  !TIT_DSP
            HL=#LCDDS+210
            break
        case 1:                            ; 7行目に設定
            HL=#LCDDS+180
            break
        case 2:                            ; 6行目に設定
            HL=#LCDDS+150
            break
        case 3:                            ; 5行目に設定
            HL=#LCDDS+120
            break
        case 4:                            ; 4行目に設定
            HL=#LCDDS+90
            break
        case 5:                            ; 3行目に設定
            HL=#LCDDS+60
            break
        case 6:                            ; 2行目に設定
            HL=#LCDDS+30
    ends
    PUSH  DE                              ; アドレス過避
    B=#10
    CALL  !DT_SET
    H=D (A)

```

```

        CALL    !TIT_DEC
        REV_ADR=H (A)
        REV_BNK=M_BANK (A)
        REV_PNT=TIT_PNT (A)
    elseif_bit(ADENDF)           ;検索データなし
        break
    endif
NEXT
ENDS
ENDIF

IF_BIT(KY_SIGF)                ;キー入力あり
CLR1    KY_SIGF
switch(KEY_CD)
case CUR_UP:                    ;反転表示行を上に移動
    if(INV_DS<#2)
        INV_DS--
        SET1    IV_CHGF        ;反転表示行変更状態に設定
    endif
    break
case CUR_DN:                    ;反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)        ;転送先アドレスの設定
        case 2:                ;3行目に設定
            HL=#LCDDS+80
            break
        case 3:                ;4行目に設定
            HL=#LCDDS+90
            break
        case 4:                ;5行目に設定
            HL=#LCDDS+120
            break
        case 5:                ;6行目に設定
            HL=#LCDDS+150
            break
        case 6:                ;7行目に設定
            HL=#LCDDS+180
            break
        case 7:                ;8行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!=#SPC) (A)    ;次の行に名前あり
            INV_DS++
            SET1    IV_CHGF    ;反転表示行変更状態に設定
        endif
    endif
    break
case CUR_R1:
    MODE4=#1
    SET1    MD_CG4F
    break
case CUR_LE:
    MODE4=#0
    SET1    MD_CG4F
    break
case CLEAR:
    MODE2=#0
    SET1    MD_CG1F
    break
case ENTER:
    switch(INV_DS)            ;検索対象行アドレスの設定
    case 2:                  ;2行目に設定
        DE=#LCDDS+30
        break
    case 3:                  ;3行目に設定
        DE=#LCDDS+60
        break
    case 4:                  ;4行目に設定
        DE=#LCDDS+90
        break
    case 5:                  ;5行目に設定
        DE=#LCDDS+120
        break
    case 6:                  ;6行目に設定
        DE=#LCDDS+150

```

```

        break
case 7:                                     ; 7行目に設定
    DE=#LCDDS+180
    break
case 8:                                     ; 8行目に設定
    DE=#LCDDS+210
ends
if([DE]!="#SPC) (A)
    MODE3=#1
    MODE4=#0
    SET1 MD_CG3F
    SET1 MD_CG4F
    HL=#TITLE1
    B=#10
    CALL !DT_SET
    FOR_ADR=#ADRSTA
    REV_ADR=#ADRSTA
    FOR_BNK=#0
    REV_BNK=#0
    FOR_PNT=#0
    REV_PNT=#0
else
    SET1 MD_CG1F
endif
break
CASE DELETE:
switch(INV_DS)                             ; 検索対象行アドレスの設定
case 2:                                     ; 2行目に設定
    DE=#LCDDS+30
    break
case 3:                                     ; 3行目に設定
    DE=#LCDDS+60
    break
case 4:                                     ; 4行目に設定
    DE=#LCDDS+90
    break
case 5:                                     ; 5行目に設定
    DE=#LCDDS+120
    break
case 6:                                     ; 6行目に設定
    DE=#LCDDS+150
    break
case 7:                                     ; 7行目に設定
    DE=#LCDDS+180
    break
case 8:                                     ; 8行目に設定
    DE=#LCDDS+210
ends
HL=#TITLE1
B=#10
CALL !DT_SET
MEMPAG=#ADRSTA
MEMBNK=#0                                  ; 現在のメモリ・アドレスを設定
B=#MEM_DAT                                 ; 検索データの設定
while(never)
    H=MEMPAG (A)
    L=#00H
    M_BANK=MEMBNK (A)
    CALL !BANKSET
    CALL !TIT_FOR                           ; タイトル(レベル1)の検索
    if_bit(ADSEAF)                          ; データあり
        CALL !DT_DEL                       ; 検索データの削除
        MEM_VOL--
    else
        break                               ; 検索データなし
    endif
endw                                         ; タイトル(レベル1)データ削除

H=#ADRSTA
L=#00H
M_BANK=#0
CALL !BANKSET
B=#TIT_DAT                                 ; タイトル(レベル1)データの検索
CALL !ADR_SEA                              ; 検索アドレス決定
if_bit(CY)                                  ; メモリ・バンク変更なし

```



```

if([HL]=B) (A) ;検索データの種類一致
  TIT_PNT=#10H
  for(i=#0;i<TIT_VOL;i++) (A)
    DE=#TITLE1
    L=TIT_PNT (A)
    C=#0
    CALL !TIT_CNP
    if_bit(!CMPBRF) ;検索文字データ一致
      CALL !TIT_DEL ;データ削除
      TIT_VOL-- ;タイトル(レベル1)データ数をデクリメント
      if(TIT_VOL=#0)
        CALL !DT_DEL
      endif
      break
    else
      CALL !TIT_INC
      if_bit(!CY)
        break
      endif
    endif
  next
else
  break
endif
endif
MODE1=#0
SET1 MD_CG3F
break
case UP:
  if(SEACHR<#'Z') ;検索文字'A'-'Y'
    SET1 MD_CG3F
    SEACHR++ ;次の文字に変更
  endif
  break
case DOWN:
  if(SEACHR>#'A') ;検索文字'B'-'Z'
    SET1 MD_CG3F
    SEACHR-- ;前の文字に変更
  endif
ends
ENDIF
RET

;*****
;メモ帳タイトル(レベル2)削除モード
;*****
ME_DT12:
  if_bit(MD_CG3F)
    CLR1 MD_CG3F
    CALL !TI2_DSP
    FOR_BNK=#0
    FOR_ADR=#ADRSTA
    REV_BNK=#0
    REV_ADR=#ADRSTA
    MODE4=#0
    SET1 MD_CG4F
  endif
  IF_BIT(MD_CG4F)
    CLR1 MD_CG4F
    SWITCH(MODE4)
    CASE 0:
      FOR(j=#0;j<#7;j++) ;1ページ表示数のカウント
        H=FOR_ADR (A)
        L=#00H
        M_BANK=FOR_BNK (A) ;現在のメモリ・アドレスを設定
        CALL !BANKSET
        B=#MEM_DAT
        CALL !TIT_FOR
        if_bit(ADSEAF)
          D=H (A) ;転送元アドレスの設定
          E=#16
          PUSH DE ;アドレス過渡
          switch(j) ;転送先アドレスの設定
          case 0:
            CALL !NEX_ADR

```

```

        CALL    ITI2_DSP
        HL=#LCDDS+30
        break
    case 1:                ; 3 行目に設定
        HL=#LCDDS+60
        break
    case 2:                ; 4 行目に設定
        HL=#LCDDS+90
        break
    case 3:                ; 5 行目に設定
        HL=#LCDDS+120
        break
    case 4:                ; 6 行目に設定
        HL=#LCDDS+150
        break
    case 5:                ; 7 行目に設定
        HL=#LCDDS+180
        break
    case 6:                ; 8 行目に設定
        HL=#LCDDS+210
    ends
    POP    DE
    CALL    !DT1LSET
    H=D (A)
    CALL    !ADR_INC
    FOR_ADR=H (A)
    FOR_BNK=M_BANK (A)
elseif_bit(ADENDF)      ; 検索データなし
    break
endif
NEXT
break
CASE 1:
FOR(j=#0;j<#7;j++)      ; 1 ページ表示数のカウント
    H=REV_ADR (A)
    L=#00H
    M_BANK=REV_BNK (A)   ; 現在のメモリ・アドレスを設定
    CALL    !BANKSET
    B=#MEM_DAT
    C=#1
    DE=#TITLE1
    CALL    !TIT_REV
    if_bit(ADSEAF)
        D=H (A)          ; 転送元アドレスの設定
        E=#16
        PUSH    DE        ; アドレス過渡
        switch(j)        ; 転送先アドレスの設定
        case 0:          ; 8 行目に設定
            CALL    !BEF_ADR
            CALL    ITI2_DSP
            HL=#LCDDS+210
            break
        case 1:          ; 7 行目に設定
            HL=#LCDDS+180
            break
        case 2:          ; 6 行目に設定
            HL=#LCDDS+150
            break
        case 3:          ; 5 行目に設定
            HL=#LCDDS+120
            break
        case 4:          ; 4 行目に設定
            HL=#LCDDS+90
            break
        case 5:          ; 3 行目に設定
            HL=#LCDDS+60
            break
        case 6:          ; 2 行目に設定
            HL=#LCDDS+30
    ends
    POP    DE
    CALL    !DT1LSET
    H=D (A)
    CALL    !ADR_DEC
    REV_ADR=H (A)

```

```

        REV_BNK=M_BANK (A)
    elseif_bit(ADENDF)           ;検索データなし
        break
    endif
NEXT
ENDS
ENDIF

IF_BIT(KY_SIGF)                 ;キー入力あり
CLR1 KY_SIGF
switch(KEY_CD)
case CUR_UP:                     ;反転表示行を上に移動
    if(INV_DS<#2)
        INV_DS--
        SET1 IV_CHGF           ;反転表示行変更状態に設定
    endif
    break
case CUR_DN:                     ;反転表示行を下に移動
    if(INV_DS<#8)
        switch(INV_DS)         ;転送先アドレスの設定
        case 2:                 ; 9 行目に設定
            HL=#LCDDS+60
            break
        case 3:                 ; 4 行目に設定
            HL=#LCDDS+90
            break
        case 4:                 ; 5 行目に設定
            HL=#LCDDS+120
            break
        case 5:                 ; 6 行目に設定
            HL=#LCDDS+150
            break
        case 6:                 ; 7 行目に設定
            HL=#LCDDS+180
            break
        case 7:                 ; 8 行目に設定
            HL=#LCDDS+210
        ends
        if([HL]!=#SPC) (A)     ;次の行に名前あり
            INV_DS++
            SET1 IV_CHGF     ;反転表示行変更状態に設定
        endif
    endif
    break
case CUR_RI:
    MODE4=#1
    SET1 MD_CG4F
    break
case CUR_LE:
    MODE4=#0
    SET1 MD_CG4F
    break
case CLEAR:
    MODE3=#0
    SET1 MD_CG4F
    break
CASE ENTER:
H=#ADRSTA
L=#00H
M_BANK=#0                       ;現在のメモリ・アドレスを設定
CALL !BANKSET
B=#MEM_DAT                       ;検索データの設定
while(forever)
CALL !TIT_FOR                     ;タイトル(レベル1)の検索
if_bit(ADSEAF)                   ;データあり
switch(INV_DS)                   ;検索対象行アドレスの設定
case 2:                           ; 2 行目に設定
    DE=#LCDDS+30
    break
case 3:                           ; 3 行目に設定
    DE=#LCDDS+60
    break
case 4:                           ; 4 行目に設定
    DE=#LCDDS+90
    break

```

```

case 5:                : 5行目に設定
  DE=#LCDDS+120
  break
case 6:                : 6行目に設定
  DE=#LCDDS+150
  break
case 7:                : 7行目に設定
  DE=#LCDDS+180
  break
case 8:                : 8行目に設定
  DE=#LCDDS+210
ends
C=#16
CALL !DTCOMP          : タイトル (レベル2) 一致
if_bit(!CMPERF)
  MEMBNK=M_BANK (A)
  MEMPAG=M (A)
  MODE2=#2           : 登録データあり→データ表示
  SET1 MD_CG2F
  break
else
  CALL !ADR_INC
  if_bit(!CY)
    if(M_BANK<=#3)
      CALL !BANKSET ;バンク変更
    else
      break
    endif
  endif
endif
else
  break
endif
endw
break
CASE DELETE:
H=#ADRSTA
L=#00H
M_BANK=#0           : 現在のメモリ・アドレスを設定
CALL !BANKSET
B=#MEM_DAT         : 検索データの設定
while(forever)
  CALL !TIT_FOR      : タイトル (レベル1) の検索
  if_bit(ADSEAF)     : データあり
  switch(INV_DS)     : 検索対象行アドレスの設定
  case 2:
    DE=#LCDDS+30
    break
  case 3:            : 3行目に設定
    DE=#LCDDS+60
    break
  case 4:            : 4行目に設定
    DE=#LCDDS+90
    break
  case 5:            : 5行目に設定
    DE=#LCDDS+120
    break
  case 6:            : 6行目に設定
    DE=#LCDDS+150
    break
  case 7:            : 7行目に設定
    DE=#LCDDS+180
    break
  case 8:            : 8行目に設定
    DE=#LCDDS+210
  ends
C=#16
CALL !DTCOMP          : タイトル (レベル2) 一致
if_bit(!CMPERF)
  CALL !DT_DEL       : 検索データの削除
  MEM_VOL--
  MODE2=#0
  SET1 MD_CG1F
  break
else

```

```

CALL !ADR_INC
if_bit(ICY)
  if(M_BANK<=#3)
    CALL !BANKSET ;バンク変更
  else
    break
  endif
endif
endif
else
  break
endif
endw
ends
ENDIF
RET

;*****
;   メモ帳1ページ削除モード
;*****
ME_DDEL:
  if_bit(MD_CG2F) ;修正データの変更
    CLR1 MD_CG2F
    M_BANK=MEMBK (A) ;メモリ・アドレスの設定
    CALL !BANKSET
    D=MEMPAG (A)
    E=#16
    HL=#LCDDS ;LCD表示アドレスの設定
    CALL !DTIPSET ;表示データの転送
    CLR1 CR_DSPF ;カーソル表示なし
    INV_DS=#0
    SET1 IV_CHGF
    SET1 LCDCGPF
  endif
  IF_BIT(KY_SIGF)
    CLR1 KY_SIGF
    switch(KEY_CD)
    case CLEAR: ;前モードに設定
      MODE2=#0
      SET1 MD_CG1F
      FOR_ADR=#ADRSTA
      REV_ADR=#ADRSTA
      FOR_BNK=#0
      REV_BNK=#0
      FOR_PNT=#0
      REV_PNT=#0
      break
    case UP:
      H=MEMPAG (A)
      L=#00H
      M_BANK=MEMBK (A) ;現在のメモリ・アドレスを設定
      CALL !BANKSET
      if(H==#ADREND) (A) ;バンク先頭アドレス
        if(M_BANK<=#3) ;バンク1-バンク3
          M_BANK++ ;バンク更新
          CALL !BANKSET
          H=#ADRSTA
          if([HL]==#MEM_DAT) (A) ;次データ=メモ帳データ
            SET1 MD_CG2F
            MEMPAG=H (A) ;アドレスの更新
            MEMBK=M_BANK (A)
          endif
        endif
      else
        H++ ;次データ参照
        if([HL]==#MEM_DAT) (A)
          SET1 MD_CG2F ;アドレスの更新
          MEMPAG=H (A)
          MEMBK=M_BANK (A)
        endif
      endif
      break
    case DOWN:
      H=MEMPAG (A)

```

```

L=#00H
M_BANK=MEMBNK (A)           ;現在のメモリ・アドレスを設定
CALL !BANKSET
if(H=#ADRSTA) (A)           ;バンク先頭アドレス
  if(M_BANK>#0)             ;バンク1-バンク3
    M_BANK--                ;バンク更新
    CALL !BANKSET
    H=#ADREND
    if([HL]==#MEM_DAT) (A)  ;次データ=メモ帳データ
      SET1 MD_CG2F
      MEMPAG=H (A)         ;アドレスの更新
      MEMBNK=M_BANK (A)
    endif
  endif
else
  H--                        ;次データ参照
  if([HL]==#MEM_DAT) (A)   ;アドレスの更新
    SET1 MD_CG2F
    MEMPAG=H (A)
    MEMBNK=M_BANK (A)
  endif
endif
break
case DELETE:
  H=MEMPAG (A)
  M_BANK=MEMBNK (A)
  CALL !BANKSET
  CALL !DT_DEL
  MEM_VOL--
  MODE2=#0
  SET1 MD_CG1F
ends
ENDIF
RET

;*****
;   タイトル検索 (アドレス前進)
;*****
TIT_FOR:
CLR1 ADSEAF
while_bit(!ADSEAF)         ;データ検索
  CALL !ADR_SEA
  if_bit(CY)               ;バンク変更
    if([HL]==B) (A)        ;検索データの種類一致
      DE=#TITLE1
      C=#1
      CALL !TIT_CMP
      if_bit(!CMPERF)      ;検索文字データ一致
        SET1 ADSEAF
        MEMPAG=H (A)      ;検索アドレスをメモリに格納
        MEMBNK=M_BANK (A)
      else
        CALL !ADR_INC
        if_bit(!CY)
          if(M_BANK<=#3)
            CALL !BANKSET ;バンク変更
          else
            break
          endif
        endif
      endif
    else
      break
    endif
  endif
else
  break
endif
else
  if(M_BANK<=#3)
    CALL !BANKSET         ;バンク変更
  else
    break
  endif
endif
endw
RET

;*****

```

```

:      タイトル検索 (アドレス後退)
;*****
TIT_REV:
  CLR1  ADSEAF
  while_bit(!ADSEAF)                ;データ検索
    CALL  !AD_SEAR
    if_bit(CY)                       ;バンク変更
      if([HL]=B) (A)                 ;検索データの種類一致
        DE=#TITLE1
        C=#1
        CALL  !TIT_CMP
        if_bit(!CMPERF)              ;検索文字データ一致
          SET1  ADSEAF
          MEMPAG=H (A)               ;検索アドレスをメモリに格納
          MEMBNK=M_BANK (A)
        else
          CALL  !ADR_DEC
          if_bit(!CY)
            if(M_BANK<=#3)
              CALL  !BANKSET         ;バンク変更
            else
              break
            endif
          endif
        else
          break
        endif
      endif
    else
      if(M_BANK!=#0)
        CALL  !BANKSET               ;バンク変更
      else
        break
      endif
    endif
  endw
  RET

;*****
:      タイトル (レベル1) 比較
;*****
TIT_CMP:
  CLR1  CMPERF
  for(i=#0;i<#10;i++)                ;タイトル(10文字)の比較データ数を設定
    A=[DE]
    if(A=[HL+C])                      ;登録データと外部メモリ・データの比較
      DE++
      C++
      ;データ一致
    else
      SET1  CMPERF                    ;データ不一致処理
      break
      ;外部メモリが大きいときCY=1
    endif
  next
  RET

;*****
:      タイトル (レベル1) 登録処理
;*****
TIT_SAV:
  X=TIT_VOL (A)                       ;データ数取り込み
  A=#0
  C=#14
  AX/=C
  ;新規ページに登録 (データ数=X0H)
  if(C=#0) (A)
    CALL  !T_SAVN                     ;タイトル(レベル1) データ登録
    ;新規ページ登録
  else
    CALL  !T_SAVO                     ;使用中のページ登録
  endif
  RET

;*****
:      新規ページのタイトル登録処理
;*****
T_SAVN:
  M_BANK=#3
  CALL  !BANKSET                       ;空き領域あり

```

```

A=!0BF00H
if(A=#VAC_DAT) ;最終アドレスのメモリ空きあり
  DE=#DTWORK ;転送用データの作成
  [DE]=#TIT_DAT (A)
  DE++
  A=#0
  for(i=#1;i<#16;i++) ;2-15バイトのデータをクリア
    [DE]=A
    DE++
  next
  HL=#TITLE1
  for(i=#0;i<#10;i++) ;タイトル(レベル1)登録
    [DE]=[HL] (A)
    HL++
    DE++
  next
  A=#0
  for(i=#10;i<#16;i++) ;2-15バイトのデータをクリア
    [DE]=A
    DE++
  next
  A=#VAC_DAT ;LCD表示データをワーク・エリアに設定
  for(i=#0;i<#224;i++)
    [DE]=A
    DE++
  next
  DE=#DTWORK
  A=#0H ;アドレス値の設定
  A<->X
  A+=MEMPAG ;検索アドレス値の加算
  HL=AX
  M_BANK=MEMBK (A)
  CALL !BANKSET
  if(H>#ADREND) (A) ;最終アドレス以上
    A=H
    A-=#ADREND ;次バンクへのアドレス補正
    H=A
    M_BANK++ ;次バンクに変更
    CALL !BANKSET
  endif
  L=#0
  CALL !DT_SAV ;データ登録
  TIT_VOL++
else
  SET1 MD.CG2F ;メモリ・アドレスに空きなし
  NODE2=#3 ;オーバフロー・モードに設定
endif
RET

;*****
; 使用中ページのタイトル登録処理
;*****
T_SAVO: ;アドレス値の設定
  A=C
  A++
  A<<=1
  A<<=1
  A<<=1
  A<<=1
  A<<=1
  A<->X
  A+=MEMPAG ;検索アドレス値の加算
  HL=AX
  if(H>#ADREND) (A) ;最終アドレス以上
    A=H
    A-=#ADREND ;次バンクへのアドレス補正
    H=A
    M_BANK++ ;次バンクに変更
    CALL !BANKSET
  endif
  DE=#TITLE1
  for(i=#0;i<#10;i++) ;タイトル(レベル1)登録
    [HL]=[DE] (A)
    HL++
    DE++
  next

```



```

        A=#00H
        for(i=#10;i<#16;i++)                ;10-15バイトのデータをクリア
            [HL]=A
            HL++
        next
        TIT_VOL++
        RET
;*****
;タイトル(レベル1)検索(アドレス前進)
;*****
TIT_FAD:
        CLR1  ADSEAF
        while_bit(!ADSEAF)                  ;データ検索
            L=#00H
            CALL  !ADR_SEA
            if_bit(CY)
                ;バンク変更
                if([HL]==B) (A)             ;検索データの種類一致
                    L=TIT_PNT (A)         ;検索アドレスの設定
                    if([HL]==SEACHR) (A)  ;検索文字データ一致
                        SET1  CY
                        SET1  ADSEAF
                        MEMPAG=M (A)
                        MEMBNK=M_BANK (A)
                        ;検索アドレスをメモリに格納
                    else
                        CALL  !TIT_INC
                        if_bit(ICY)
                            break
                        endif
                    endif
                else
                    break
                endif
            else
                if(M_BANK<=#3)
                    CALL  !BANKSET          ;バンク変更
                else
                    break
                endif
            endif
        endw
        RET
;*****
;タイトル(レベル1)検索(アドレス後退)
;*****
TIT_RAD:
        CLR1  ADSEAF
        while_bit(!ADSEAF)                  ;データ検索
            L=#00H
            CALL  !AD_SEAR
            if_bit(CY)
                ;バンク変更
                if([HL]==B) (A)             ;検索データの種類一致
                    L=TIT_PNT (A)         ;検索アドレスの設定
                    if([HL]==SEACHR) (A)  ;検索文字データ一致
                        SET1  CY
                        SET1  ADSEAF
                        MEMPAG=M (A)
                        MEMBNK=M_BANK (A)
                        ;検索アドレスをメモリに格納
                    else
                        CALL  !TIT_DEC
                        if_bit(ICY)
                            break
                        endif
                    endif
                else
                    break
                endif
            else
                if(M_BANK<=#3)
                    CALL  !BANKSET          ;バンク変更
                else
                    break
                endif
            endif
        endw

```

```

RET
;*****
; タイトル (レベル1) アドレス前進
;*****
TIT_INC:
    if(TIT_PNT!="#0E0H)                ;1 ページ最終アドレス
        TIT_PNT+="#10H                ;次の行アドレスに設定
        SBT1    CY
    else
        CALL    !ADR_INC                ;次ページに設定
        if_bit(!CY)
            if(M_BANK<="#3)
                CALL    !BANKSET        ;バンク変更
                SET1    CY
            else
                CLR1    CY
            endif
        endif
        TIT_PNT="#10H                ;ページ開始行の設定
    endif
    RET
;*****
; タイトル (レベル1) アドレス後退
;*****
TIT_DEC:
    if(TIT_PNT!="#10H)                ;1 ページ最終アドレス
        TIT_PNT-="#10H                ;次の行アドレスに設定
        SBT1    CY
    else
        CALL    !ADR_DEC                ;次ページに設定
        if_bit(!CY)
            if(M_BANK<="#3)
                CALL    !BANKSET        ;バンク変更
                SET1    CY
            else
                CLR1    CY
            endif
        endif
        TIT_PNT="#0E0H                ;ページ開始行の設定
    endif
    RET
;*****
; タイトル (レベル1) データ削除
;*****
TIT_DEL:
    while(forever)
        if(L1="#0E0H) (A)                ;1 頁内の最終行
            [HL]=[HL+10H] (A)            ;次の行データ (+10H) 参照
            L++
        ELSEIF(H="#ADREND) (A)            ;バンク内の最終アドレス
            if(M_BANK<="#3)
                M_BANK++                ;最終バンクでない
                M_BANK++                ;次のバンクに設定
                CALL    !BANKSET
                D=#ADRSTA                ;先頭アドレスの設定
                E=#0
                if([DE]=#TIT_DAT) (A)    ;先頭データがタイトル (レベル1) データ
                    HL=#DTWORK          ;ワーク・エリアの設定
                    for(i=#0;i<#10H;i++) ;1 行分のデータをワークに転送
                        [HL]=[DE] (A)
                        HL++
                        DE++
                    next
                    M_BANK--            ;前のバンクに設定
                    CALL    !BANKSET
                    D=#ADREND          ;最終アドレスに設定
                    E=#0E0H
                    for(i=#0;i<#10H;i++) ;最終行にデータ登録
                        [DE]=[HL] (A)
                        DE++
                        HL++
                    next
                    H=#ADRSTA          ;次バンクの先頭ページに設定
                    L=#0
                    M_BANK++

```

```

CALL    !BANKSET
if([HL+20H]==#VAC_DAT) (A) ;タイトル・データなし
CALL    !DT_DEL           ;データ削除
MEM_VOL--
break
else
L=#10H
endif
else
M_BANK-- ;前のメモリ・バンクに設定
CALL    !BANKSET
H=#ADREND ;バンク内の最終アドレスに設定
L=#0E0H
A=#VAC_DAT ;データなし状態に設定
for(i=#0;i<#20H;i++) ;最終行をデータなしに設定
[HL]=A
HL++
next
break
endif
endif
else
A=#VAC_DAT ;データなし状態に設定
for(i=#0;i<#20H;i++) ;最終行をデータなしに設定
[HL]=A
HL++
next
break
endif
endif
else
if([HL+20H]!=#FIT_DAT) (A) ;次ページはタイトル・データでない
A=#VAC_DAT
for(i=#0;i<#20H;i++) ;最終行を空きデータに設定
[HL]=A
HL++
next
break
endif
else
for(i=#0;i<#10H;i++) ;次ページ先頭データを転送
[HL]=[HL+20H] (A)
HL++
next
H++ ;次ページ先頭アドレスに設定
L=#00H
if([HL+20H]==#VAC_DAT) (A) ;2行目以降タイトル・データなし
CALL    !DT_DEL           ;データ削除
MEM_VOL--
break
else
L=#10H ;タイトル・データ先頭に設定
endif
endif
endif
endw
RET

;*****
;   タイトル (レベル1) 表示設定
;*****
TI1_DSP:
CALL    !LCD_DSC           ;LCD表示エリアをクリア
HL=#LCDDS
DE=#T_DAT                 ;1行目の表示データを設定
CALL    !DTILSET
!LCDDS+11=SEACHR (A)      ;検索文字の表示
INV_DS=#2
SET1   IV_CHGF
SET1   LCDCHGF
RET

;*****
;   タイトル (レベル2) 表示設定
;*****
TI2_DSP:
CALL    !LCD_DSC           ;LCD表示エリアをクリア
HL=#LCDDS

```

```
DE=#TITLE1           ;1行目の表示データを設定
B=#10
CALL  !DT_SET
INV_DS=#2
SET1  IV_CHGF
SET1  LCDCHGF
RET
END
```

4.8 世界時計モード

世界時計モードでは、東京の時間を基準にして、世界都市14箇所の時間を算出し、表示を行っています。世界時計には、時計表示処理と、都市選択処理の2つがあります。

4.8.1 モード選択条件

世界時計モードの遷移状態の条件を表4-9に示します。

表4-9 世界時計モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
5	0	-	-	-	世界時計モード選択
	1	-	-	-	時計表示モード
	2	0	-	-	世界都市（1ページ目）選択モード
		1	-	-	世界都市（2ページ目）選択モード

備考 - : don't care

4.8.2 時計表示処理

時計表示処理では、東京の時間に対して各都市ごとに時差データを持たせ、そのデータを使って選択された都市の時間を算出しています。時差データの構成を図4-19に、各都市の時差データを表4-10に示します。CLRキー入力により世界時計モード選択に戻します（MODE1=0）。

図4-19 時差データの構成

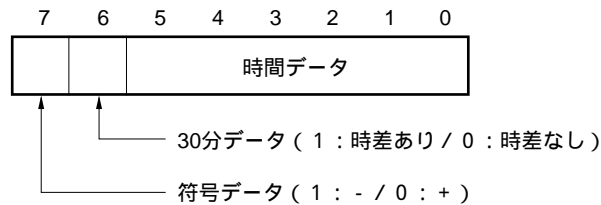


表4-10 世界時計データ

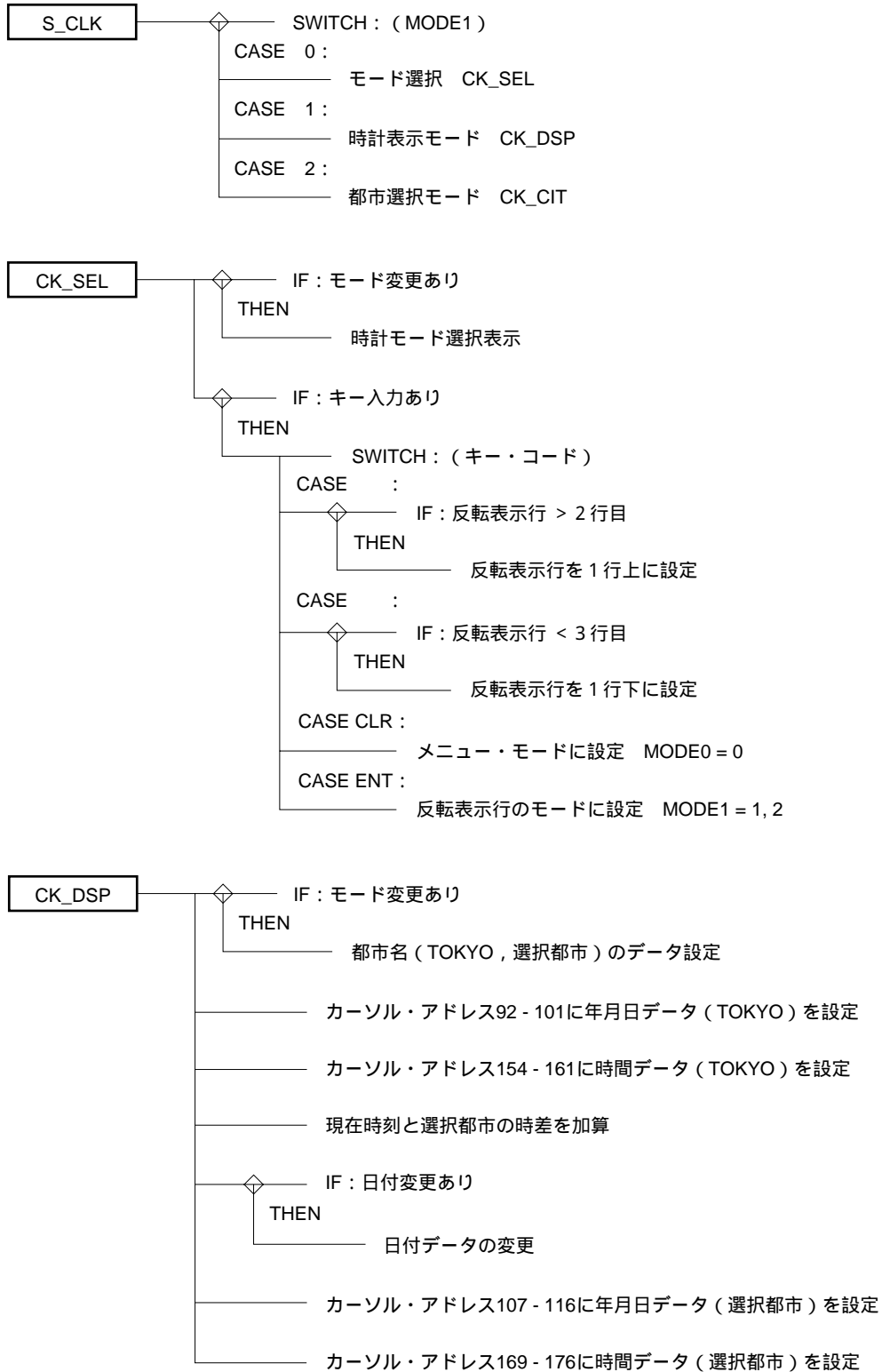
ページ	都市名	都市データ	時 差	時差データ
1	NEW YORK	1	- 14 : 00 : 00	10001110B
	LOS ANGELES	2	- 17 : 00 : 00	10010001B
	BRASILIA	3	- 12 : 00 : 00	10001100B
	LIMA	4	- 14 : 00 : 00	10001110B
	LONDON	5	- 9 : 00 : 00	10001001B
	ROME	6	- 8 : 00 : 00	10001000B
	CAIRO	7	- 7 : 00 : 00	10000111B
2	MOSCOW	8	- 6 : 00 : 00	10000110B
	HONOLULU	9	- 19 : 00 : 00	10010011B
	SYDNEY	10	+ 1 : 00 : 00	00000001B
	PEKING	11	- 1 : 00 : 00	10000001B
	HONG KONG	12	- 1 : 00 : 00	10000001B
	NEW DELHI	13	- 3 : 30 : 00	11000011B
	KUWAIT	14	- 6 : 00 : 00	10000110B

4.8.3 都市選択処理

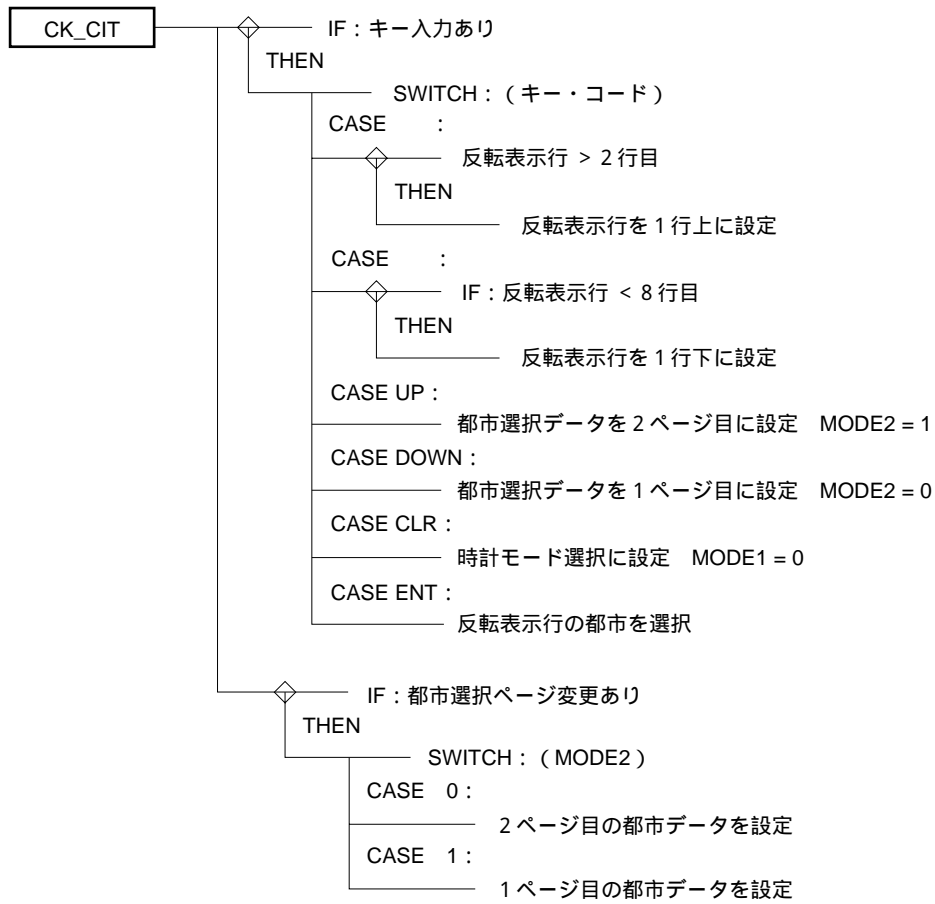
都市選択では、UP/DOWNキー入力により、世界都市を2ページに分割し、7箇所ずつ表示しています。そして、ENTキー入力により、表示中のページ・データ (MODE2) と、反転表示行 (INV_DS) から都市データを決定し、選択都市の設定 (CIT_DAT) をします。

4.8.4 SPDチャート

世界時計モード



世界時計モード



4.8.5 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, NR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
EXTRN  SECD, WIND, HOUD, DAYD, MOND, YEAD, M_BANK, MEMPAG, MEMBNK, BACPAG, BACBNK
EXTRN  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
EXTRN  VAC_DAT, ADM_DAT, ADC_DAT, MEN_DAT, SCA_DAT, HOL_DAT
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DTILSET, DTIPSET, LCD_C, LCD_D, DT_SAV, STA_SEA
EXTRN  SAVADR, DTCOMP, DT_DEL, DEL_IL, BANKSET, ADR_SEA, AD_SEAR, ADR_INC, ADR_DEC, DT_SET
EXTRN  SEA_FOR, SEA_REV, SEA_FIL, SEA_RIL, YEADSP, MONDSP, CAL_LEA, CAL_MOR
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, IV_CHGF, COMM_F, DS_ERAF
EXTBIT  CR_DSPF, ADENDF, ADSEAF, CMPERF, LCDCHGF

PUBLIC  S_CLK, CIT_DAT, ASC_CNV

$      INCLUDE (LCDCMD.INC)
$      INCLUDE (LCDT_EX.INC)

$LIST

      DSEG      SADDR
CIT_DAT:DS      1                                ;表示都市データ

;*****
;      世界時計分秒処理
;*****
CLKSEG      CSEG
S_CLK:
      switch(NODE1)
      case 0:
            CALL  ICK_SEL                                ;モード選択
            break
      case 1:
            CALL  ICK_DSP                                ;時計表示モード
            break
      case 2:
            CALL  ICK_CIT                                ;都市選択モード
      ends
      RET

;*****
;      世界時計モード選択処理
;*****
CK_SEL:
      if_bit(MD_CG1F)
            CLR1  MD_CG1F
            HL=#LCDDS                                ;メニュー表示データの設定
            DE=#CLOCK                                ;LCD 1行目のデータを設定
            CALL  !DTILSET

            DE=#DISPLAY                                ;LCD 2行目のデータを設定
            CALL  !DTILSET

            DE=#CIT_SEL                                ;LCD 3行目のデータを設定
            CALL  !DTILSET

            DE=#CLR1LD                                ;LCD 4行目のデータを設定
            CALL  !DTILSET

            DE=#CLR1LD                                ;LCD 5行目のデータを設定
            CALL  !DTILSET

            DE=#CLR1LD                                ;LCD 6行目のデータを設定
            CALL  !DTILSET

            DE=#CLR1LD                                ;LCD 7行目のデータを設定
            CALL  !DTILSET

```

```

DE=#CLR1LD ;LCD 8行目のデータを設定
CALL !DT1LSET

MODE2=#0
MODE3=#0
MODE4=#0
SET1 MD_CG2F
SET1 MD_CG3F
SET1 MD_CG4F
CLR1 CR_DSPF
SET1 IV_CHGF ;反転表示行を2行目に設定
INV_DS=#2
SET1 LCDCHGF
endif

if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD) ;キー・コード分岐
case CUR_UP:
    if(INV_DS<#2)
        INV_DS--
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case CUR_DN: ;反転表示行を下に移動
    if(INV_DS<#3)
        INV_DS++
        SET1 IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case CLEAR: ;前画面モードに変更
    MODE0=#0
    SET1 MD_CG0F
    break
case ENTER: ;反転表示行のモード実行
    A=INV_DS
    A--
    MODE1=A
    SET1 MD_CG1F ;モード変更状態に設定
ends
endif
RET

;*****
; 世界時計表示処理
;*****
CK_DSP:
if_bit(MD_CG1F)
CLR1 MD_CG1F
INV_DS=#0 ;反転表示削除
SET1 IV_CHGF
CALL !LCD_DSC ;LCD表示クリア
!LCDDS+30+5=#'T' (A) ;T O K Y Oを表示
!LCDDS+30+6=#'O' (A)
!LCDDS+30+7=#'K' (A)
!LCDDS+30+8=#'Y' (A)
!LCDDS+30+9=#'O' (A)

A=CIT_DAT ;都市選択データ
A--
A+=A ;都市データをアドレス・データに変換
A+=A ;1データ16バイト
A+=A
X=A
A=#0
AX+=#CITD_T ;都市表示データ先順アドレスを加算
DE=AX
HL=#LCDDS+30+17
for(i=#0;i<#11;i++) ;都市選択データを表示
    [HL]=[DE] (A)
    DE++
    HL++
next
SET1 LCDCHGF

```

```

endif
HL=#LCDDS+90+2
X=YEAD (A) ;年データの取り込み
CALL !YEADSP ;年データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=MOND (A) ;月データの取り込み
CALL !MONDSP ;月データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=DAYD (A) ;日データの取り込み
CALL !MONDSP ;日データ表示

HL=#LCDDS+150+4
A=HOUD+1 ;時(上位)データ設定
CALL !ASC_CNV
A=HOUD ;時(下位)データ設定
CALL !ASC_CNV
[HL]=#COLON (A)
HL++
A=MIND+1 ;分(上位)データ設定
CALL !ASC_CNV
A=MIND ;分(下位)データ設定
CALL !ASC_CNV
[HL]=#COLON (A)
HL++
A=SECD+1 ;秒(上位)データ設定
CALL !ASC_CNV
A=SECD ;秒(下位)データ設定
CALL !ASC_CNV

HL=#LCDDS+150+23
A=MIND ;分(下位)データ設定
CALL !ASC_CNV
[HL]=#COLON (A)
HL++
A=SECD+1 ;秒(上位)データ設定
CALL !ASC_CNV
A=SECD ;秒(下位)データ設定
CALL !ASC_CNV

!LCDDS+150+21=#COLON (A)
B=CIT_DAT (A) ;都市選択データの取り込み
B--
HL=#TIM_DIF ;時差データ・テーブルの設定
B=[HL+B] (A) ;選択都市の時差データの取り込み
IF_BIT(1A.7) ;時差の値>0
  if_bit(A.6) ;30分データあり
  A=#3 ;分(上位)に3を加算
  A+=MIND+1
  if(A>#6) ;桁上げあり
  A=#6 ;分データの補正
  X=#1 ;時データ加算値の設定
  else
  X=#0 ;時データ加算値の設定
  endif
else
  A=MIND+1
  X=#0 ;時データ加算値の設定
endif
endif
HL=#LCDDS+150+22
CALL !ASC_CNV ;分(上位)データ設定
A=B
A&=#0011111B ;時差時間データの取り込み
X+=A
A=HOUD+1 ;時(上位)データの取り込み
A+=A ;2倍
A+=A ;4倍
A+=HOUD+1 ;5倍
A+=A ;10倍
A+=HOUD ;現在の時(下位)データを加算
A+=X ;時差データを加算
if(A>#24) ;日付変更あり
  A=#24 ;時データの補正
  E=#1 ;日付変更ありに設定

```

```

else
    E=#0 ;日付変更なしに設定
endif
X=A ;表示用データに変換
A=#0
C=#10 ;10進補正
AX/-C ;結果=時(上位), 余り=時(下位)
HL=#LCDDS+150+19
A=X ;時(上位)データの設定
CALL !ASC_CNV
A=C ;時(下位)データの設定
CALL !ASC_CNV

if(EI=#0) (A)
    C=DAYD (A)
    C++ ;日データ・インクリメント
    A=YEAD ;現在月の最終日を参照
    A&=#00000011B
    if_bit(Z) ;うるう年(下位2ビット=0)?
        HL=#CAL_LEA ;うるう年データの参照
    else
        HL=#CAL_NOR ;平年データの参照
    endif
    B=MOND (A)

    B--

    A=[HL+B]
    if(A<C) ;最終日オーバ
        C=#1 ;現在日付を1日に設定
        D=MOND (A)
        D++ ;月データ・インクリメント
        if(D>#12) (A) ;年変更(12月オーバ)
            D=#1 ;現在月を1月に設定
            E=YEAD (A)
            E++ ;年データ・インクリメント
        else
            E=YEAD (A)
        endif
    else
        D=MOND (A) ;表示月データの設定
        E=YEAD (A) ;表示年データの設定
    endif
else
    C=DAYD (A) ;表示日データの設定
    D=MOND (A) ;表示月データの設定
    E=YEAD (A) ;表示年データの設定
endif
ELSE
    if_bit(A.6) ;30分データあり
    A=#100H-3 ;分(上位)から3を減算
    A+=MIND+1
    if_bit(ICY) ;桁上がりあり
        X=#1 ;時データ減算値の設定
        A+=#6 ;分データの補正
    else
        X=#0 ;時データ減算値の設定
    endif
else
    A=MIND+1
    X=#0 ;時データ減算値の設定
endif
HL=#LCDDS+150+22
CALL !ASC_CNV ;分(上位)データ設定
A=B
A&=#00111111B ;時差時間データの取り込み
A+=X
X=A ;分桁上りを加算
A=HOUD+1 ;時(上位)データの取り込み
A+=A ;2倍
A+=A ;4倍
A+=HOUD+1 ;5倍
A+=A ;10倍

```

```

A+=HOUD ;現在の時(下位)データを加算
A-=X ;時差データを減算
if_bit(CY) ;日付変更あり
    A+=#24 ;時データの補正
    E=#1 ;日付変更ありに設定
else
    E=#0 ;日付変更なしに設定
endif
X=A ;表示用データに変換
A=#0
C=#10 ;10進補正
AX/=C ;結果=時(上位), 余り=時(下位)
HL=#LCDDS+150+19
A=X ;時(上位)データの設定
CALL !ASC_CNV
A=C ;時(下位)データの設定
CALL !ASC_CNV

if(E1=#0) (A) ;日付変更あり
    C=DAYD (A) ;日データ・デックリメント
    C-- ;月変更
    if_bit(Z) ;現在月=1月
        if(WOND==#1) ;12月に設定
            B=#12 ;表示月データの設定
            A=YEAD ;現在月の最終日を参照
            A-- ;表示年データの設定
            E=A ;表示年データの設定
            A&=#00000011B
        else
            B=WOND (A) ;現在月データの取り込み
            B-- ;前月の設定
            D=B (A) ;表示月データの設定
            A=YEAD ;現在月の最終日を参照
            E=A ;表示年データの設定
            A&=#00000011B
        endif
        B--
        if(A==#0) ;うるう年(下位2ビット=0)?
            HL=#CAL_LEA ;うるう年データの参照
        else
            HL=#CAL_NOR ;平年データの参照
        endif
        C=[HL+B] (A) ;表示日データの設定
    else
        D=WOND (A) ;表示月データの設定
        E=YEAD (A) ;表示年データの設定
    endif
else
    C=DAYD (A) ;表示日データの設定
    D=WOND (A) ;表示月データの設定
    E=YEAD (A) ;表示年データの設定
endif
endif

HL=#LCDDS+90+17
X=E (A) ;年データの取り込み
CALL !YEADSP ;年データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=D (A) ;月データの取り込み
CALL !MONDSP ;月データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=C (A) ;日データの取り込み
CALL !MONDSP ;日データ表示

if_bit(KY_SIGF)
    CLR1 KY_SIGF
    switch(KEY_CD)
    case CLEAR: ;前モードに設定
        MODE1=#0
        SET1 MD_CG1F
    ends
endif

```

```

RET

;*****
;      世界時計表示処理
;*****
CK_CIT:
  if_bit(MD_CG2F)
    CLR1  MD_CG2F
    INV_DS=#2
    SET1  IV_CHGF
    HL=#LCDDS
    DE=#CIT_SEL
    CALL  !DTILSET
    switch(MODE2)
    case 0:
      DE=#NEWYORK
      CALL  !DTILSET           ; 2行目のデータを設定

      DE=#LOSANG
      CALL  !DTILSET           ; 3行目のデータを設定

      DE=#BRASIL
      CALL  !DTILSET           ; 4行目のデータを設定

      DE=#LIMA
      CALL  !DTILSET           ; 5行目のデータを設定

      DE=#LONDON
      CALL  !DTILSET           ; 6行目のデータを設定

      DE=#ROME
      CALL  !DTILSET           ; 7行目のデータを設定

      DE=#CAIRO
      CALL  !DTILSET           ; 8行目のデータを設定
      break
    case 1:
      DE=#MOSCOW
      CALL  !DTILSET           ; 2行目のデータを設定

      DE=#HONOLU
      CALL  !DTILSET           ; 3行目のデータを設定

      DE=#SYDNEY
      CALL  !DTILSET           ; 4行目のデータを設定

      DE=#PEKING
      CALL  !DTILSET           ; 5行目のデータを設定

      DE=#HONG
      CALL  !DTILSET           ; 6行目のデータを設定

      DE=#NEWDEL
      CALL  !DTILSET           ; 7行目のデータを設定

      DE=#KUWAIT
      CALL  !DTILSET           ; 8行目のデータを設定
    ends
    SET1  LCDCHGF
  endif

  if_bit(KY_SIGF)
    CLR1  KY_SIGF
    switch(KEY_CD)
    case CUR_UP:           ; 反転表示行を上に移動
      if(INV_DS>#2)
        INV_DS--
        SET1  IV_CHGF       ; 反転表示行変更状態に設定
      endif
      break
    case CUR_DN:           ; 反転表示行を下に移動
      if(INV_DS<#8)
        INV_DS++
        SET1  IV_CHGF       ; 反転表示行変更状態に設定
      endif
    ends
  endif

```

```

        endif
        break
    case UP:
        if(MODE2=#0)                ;検索文字'A'-'Y'
            MODE2=#1                ;モード変更
            SET1 MD_CG2F
        endif
        break
    case DOWN:
        if(MODE2=#1)                ;検索文字'B'-'Z'
            MODE2=#0                ;モード変更
            SET1 MD_CG2F
        endif
        break
    case CLEAR:
        MODE1=#0                    ;前面モードに変更
        SET1 MD_CG1F
        break
    case ENTER:
        ;反転表示行のモード実行
        A=INV_DS
        A--
        if(MODE2=#0)                ;1ページのデータ
            A+#0                    ;選択都市データ範囲1-7
        else
            A+#7                    ;選択都市データ範囲8-14
        endif
        CIT_DAT=A                  ;選択都市データの設定
        MODE1=#0
        SET1 MD_CG1F                ;モード変更状態に設定
        SET1 IV_CHGF
        INV_DS=#0
    ends
endif
RET

;*****
;          表示コード変換
;*****
ASC_CNV:
    A=#30H
    [HL]=A
    HL++
    RET

;*****
;          世界時計時差データ
;*****
TIM_DIF:
    ;データ構成 ビット7:符号(1=-/0=+)
    ;ビット6:時差30分データ(1=時差30分)
    ;ビット0-5:時差時間データ
    DB 10001110B ;NEW YORK -14時間
    DB 10010001B ;LOS ANGELES -17時間
    DB 10001100B ;BRASILIA -12時間
    DB 10001110B ;LIMA -14時間
    DB 10001001B ;LONDON -9時間
    DB 10001000B ;ROME -8時間
    DB 10000111B ;CAIRO -7時間

    DB 10000110B ;MOSCOW -6時間
    DB 10010011B ;HONOLULU -19時間
    DB 00000001B ;SYDNEY +1時間
    DB 10000001B ;PEKING -1時間
    DB 10000001B ;HONG KONG -1時間
    DB 11000011B ;NEW DELHI -3時間30分
    DB 10000110B ;KUWAIT -6時間

;*****
;          世界時計都市表示データ
;*****
CITD_T:
    DB SPC,'N','E','W',SPC,'Y','O','R' ;NEW YORK
    DB 'K',SPC,SPC,SPC,SPC,SPC,SPC,SPC
    DB 'L','O','S',SPC,'A','N','G','E' ;LOS ANGELES
    DB 'L','E','S',SPC,SPC,SPC,SPC,SPC

```

```
DB   SPC, 'B', 'R', 'A', 'S', 'I', 'L', 'I'   ; B R A S I L I A
DB   'A', SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, SPC, 'L', 'I', 'M', 'A', SPC   ; L I M A
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, 'L', 'O', 'N', 'D', 'O', 'N'   ; L O N D O N
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, SPC, 'R', 'O', 'M', 'E', SPC   ; R O M E
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, SPC, 'C', 'A', 'I', 'R', 'O'   ; C A I R O
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, 'M', 'O', 'S', 'C', 'O', 'W'   ; M O S C O W
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, 'H', 'O', 'N', 'O', 'L', 'U', 'L', 'U' ; H O N O L U L U
DB   'U', SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, 'S', 'Y', 'D', 'N', 'E', 'Y'   ; S Y D N E Y
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, 'P', 'E', 'K', 'I', 'N', 'G'   ; P E K I N G
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, 'H', 'O', 'N', 'G', SPC, 'K', 'O'   ; H O N G   K O N G
DB   'N', 'G', SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, 'N', 'E', 'W', SPC, 'D', 'E', 'L', 'I' ; N E W   D E L H I
DB   'H', 'I', SPC, SPC, SPC, SPC, SPC, SPC

DB   SPC, SPC, 'K', 'U', 'W', 'A', 'I', 'T'   ; K U W A I T
DB   SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
END
```


4.9 初期設定モード

初期設定モードでは、日付の設定と、時刻の設定を行っています。リセット・スタートのときは、1992年7月1日00時00分00秒に設定しています。

4.9.1 モード選択条件

初期設定モードの遷移状態の条件を表4-11に示します。

表4-11 初期設定モードのMODE0-MODE4の状態

MODE0	MODE1	MODE2	MODE3	MODE4	遷移モード
6	0	-	-	-	初期設定モード選択
	1	0	-	-	年設定モード
		1	-	-	月設定モード
		2	-	-	日設定モード
	2	0	-	-	時設定モード
		1	-	-	分設定モード
		2	-	-	秒設定モード

備考 - : don't care

4.9.2 日付の設定

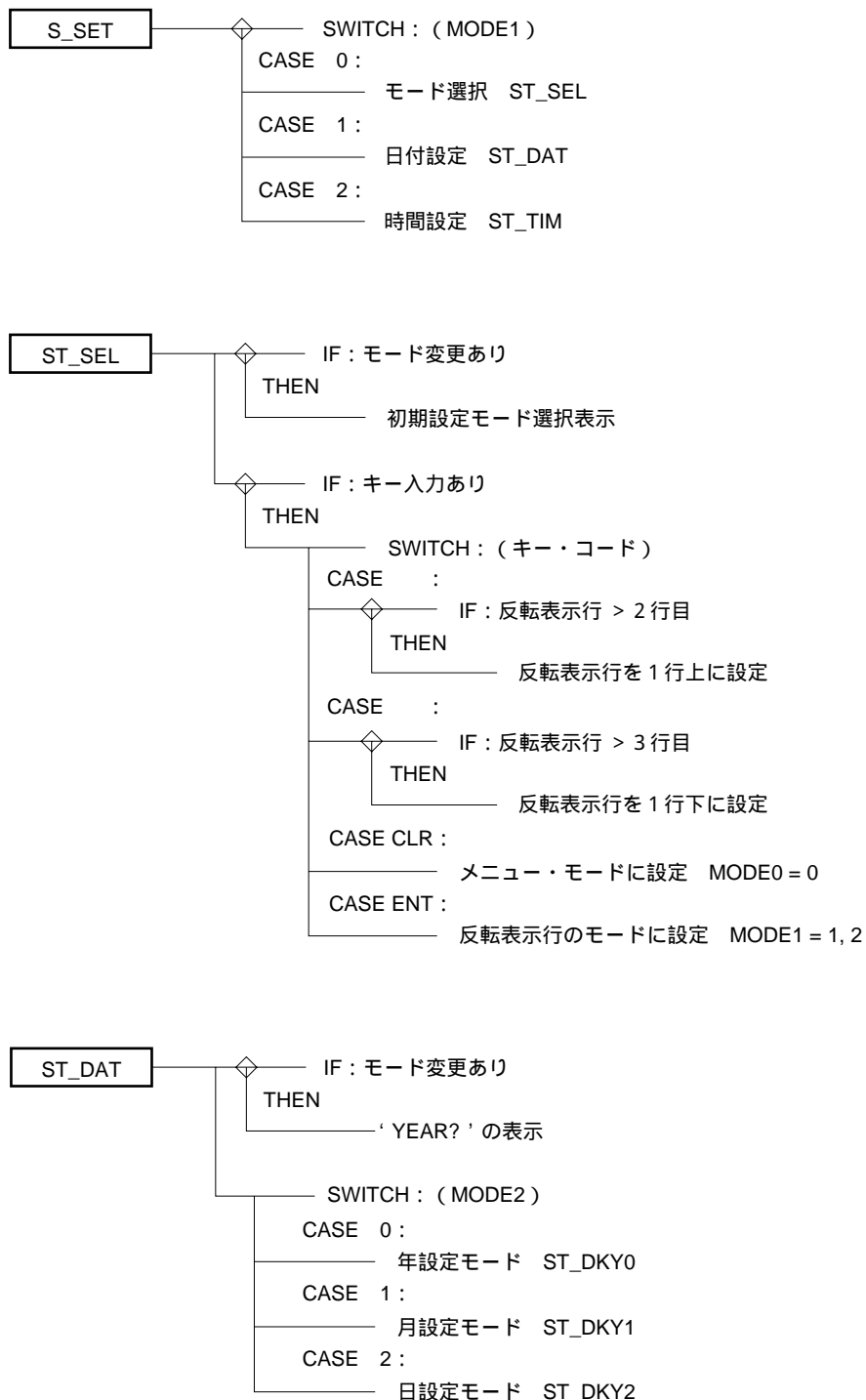
日付設定モードでは、ENTキー入力によりMODE2の値を更新し、CLRキー入力のときは前モードに戻すためMODE1を0に設定します。日付データは、数字データのキー入力により、カーソルのあるアドレス（CUR_ADR）に数字データを書き込みます。年 月 日の順にデータを入力します。日データ入力後のENTキーによりデータが有効（YEAD, MOND, DAYDにデータを設定）になります。

4.9.3 時刻の設定

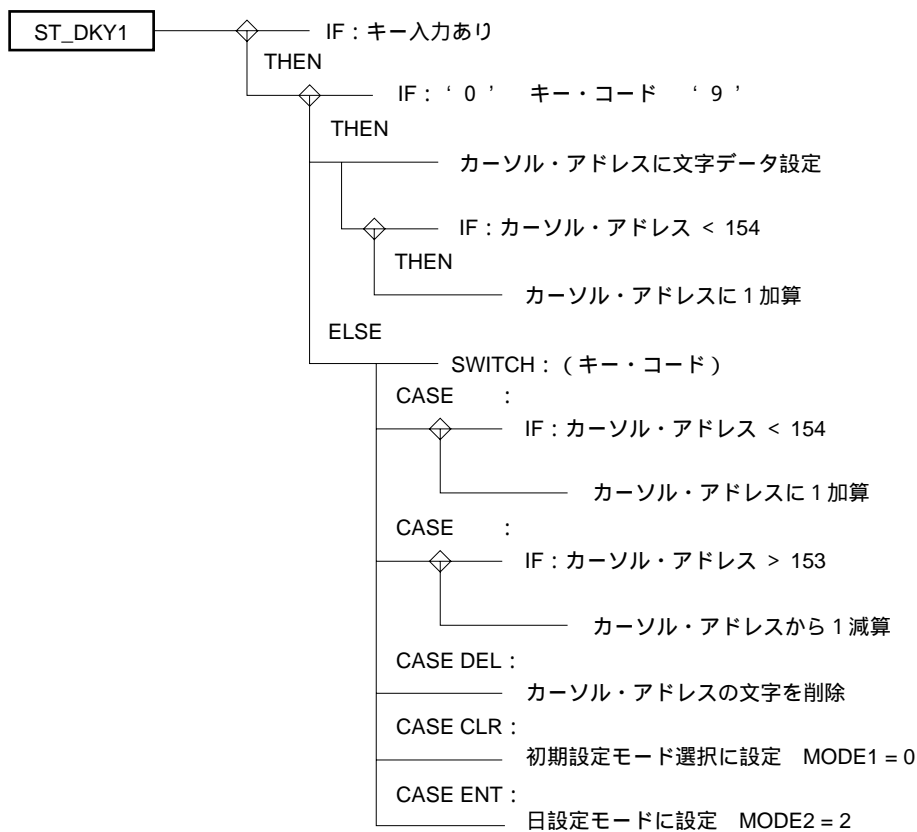
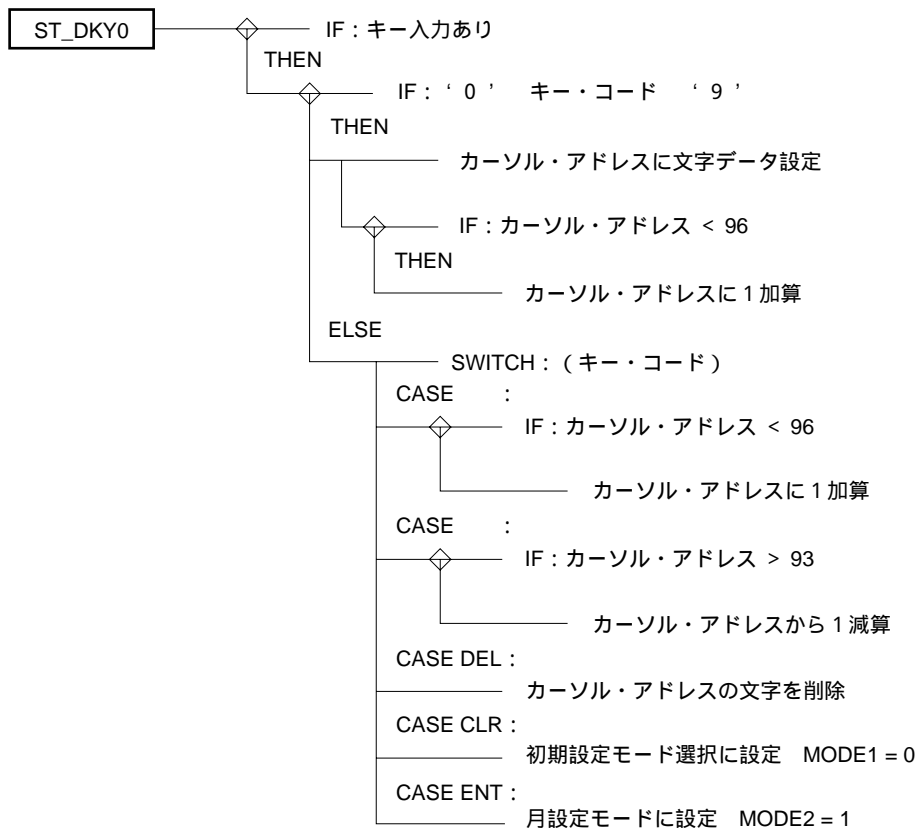
時刻設定モードでは、ENTキー入力によりMODE2の値を更新し、CLRキー入力のときは前モードに戻すためMODE1を0に設定します。時刻データは、数字データのキー入力により、カーソルのあるアドレス（CUR_ADR）に数字データを書き込みます。時 分 秒の順にデータを入力します。秒データ入力後のENTキーによりデータが有効（HOUD, MIND, SECDにデータを設定）になります。

4.9.4 SPDチャート

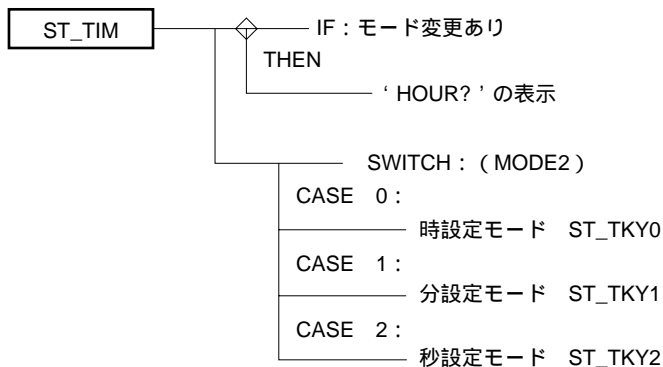
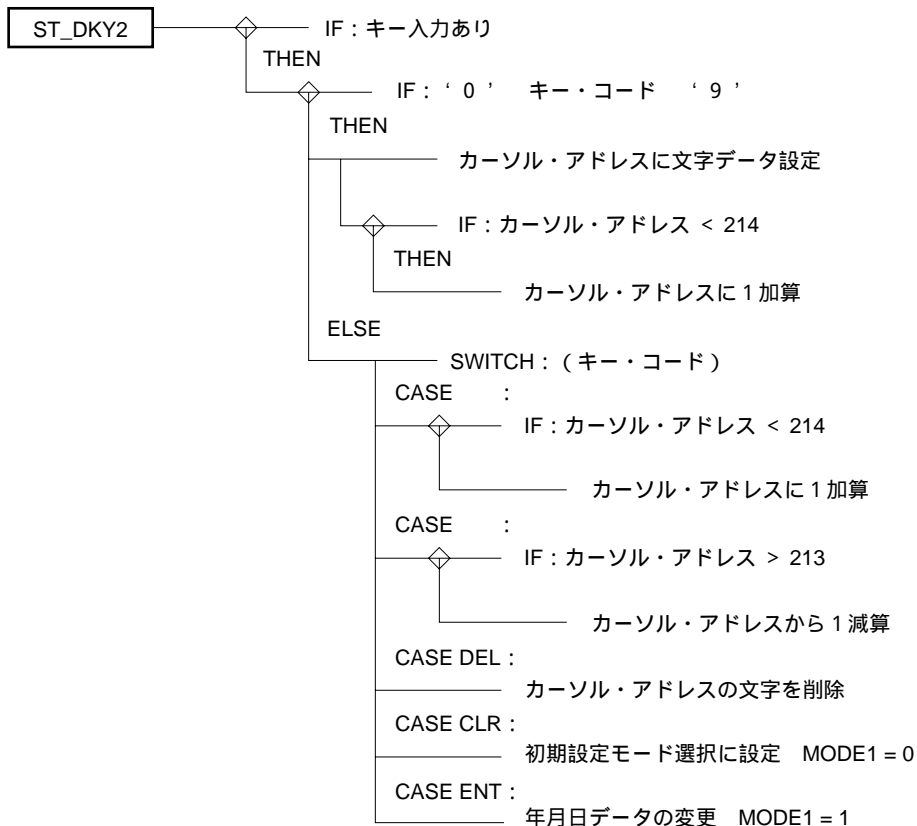
初期設定モード



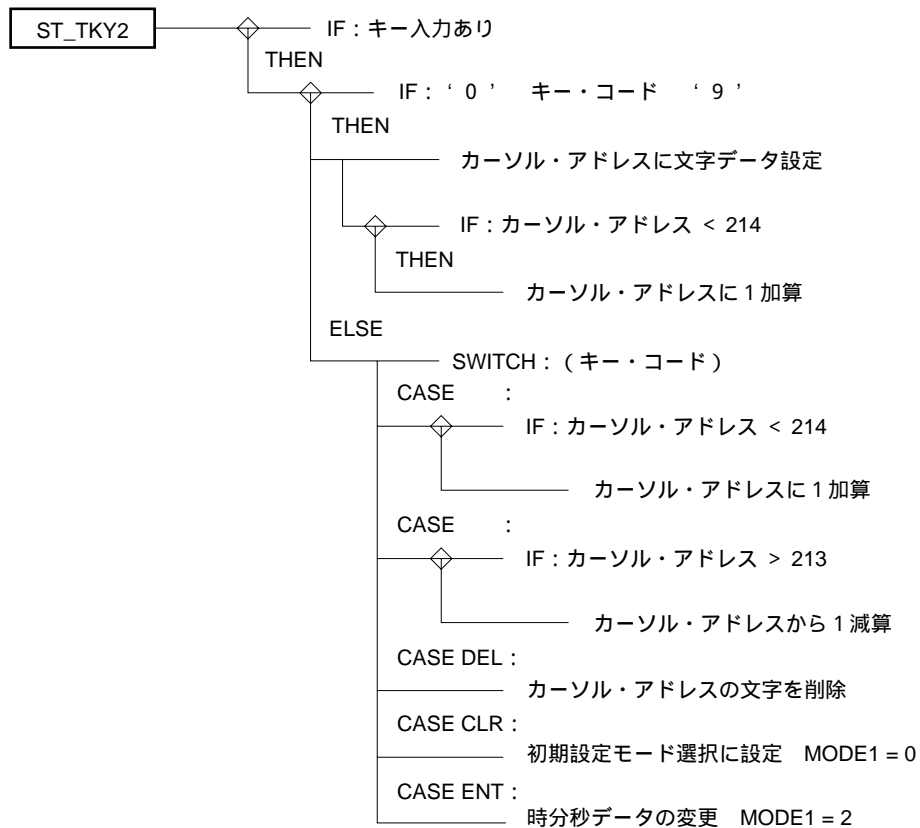
初期設定



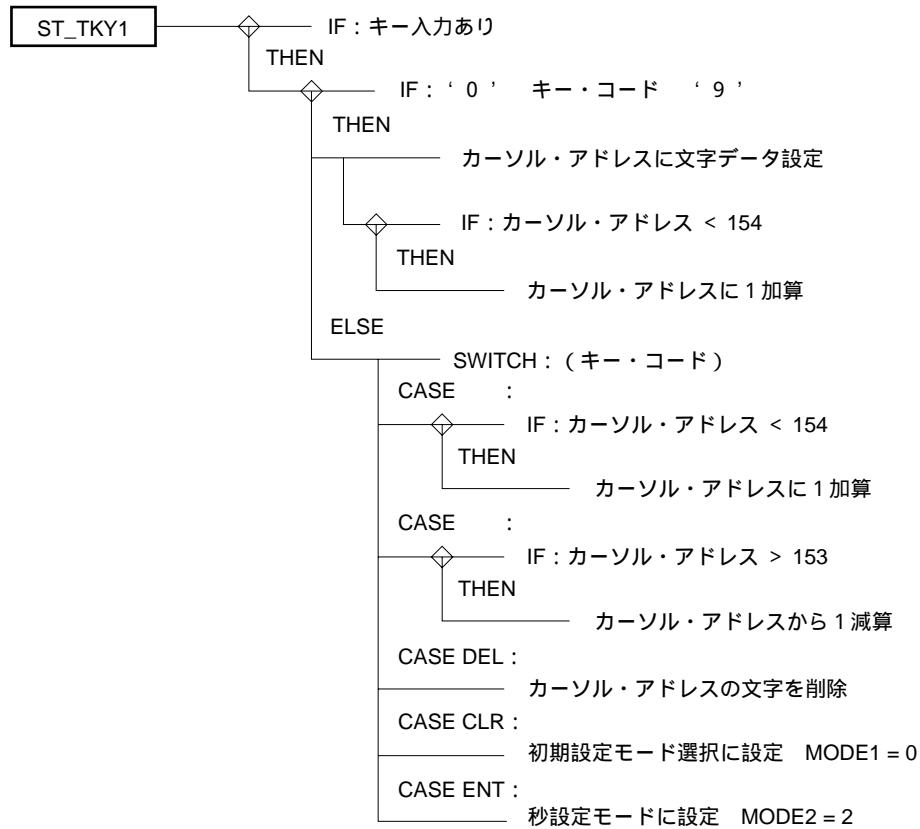
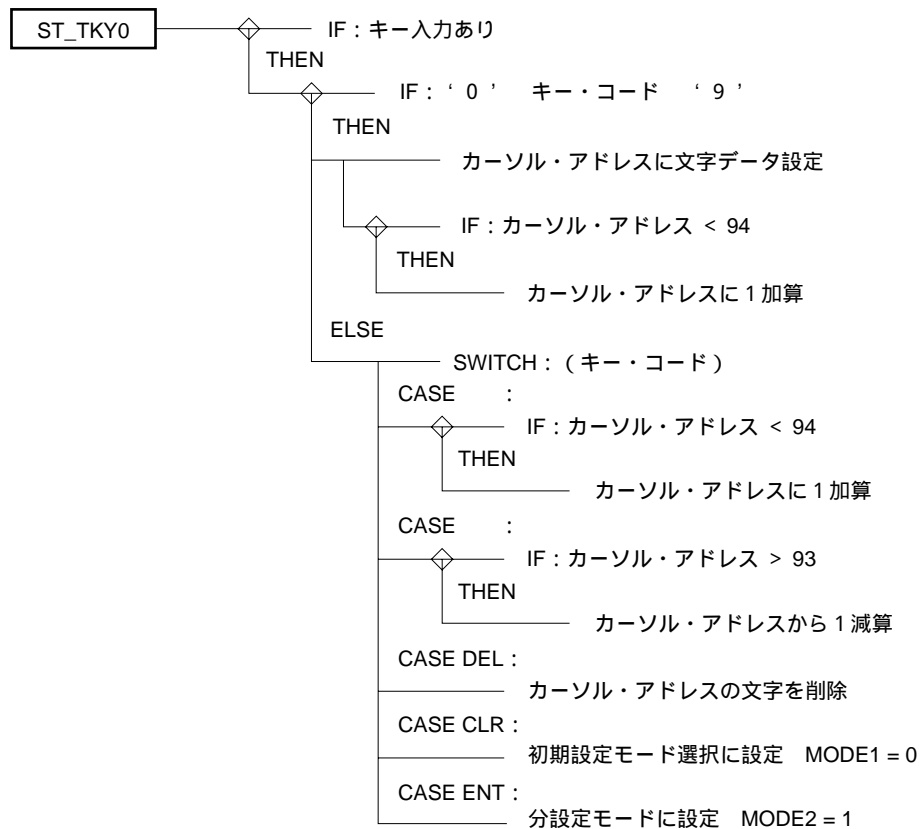
初期設定



初期設定



初期設定



4.9.5 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  MODE0, MODE1, MODE2, MODE3, MODE4, CUR_ADR, KEY_CD, KEYDAT, KEYWK, INV_DS
EXTRN  SECD, MIND, HOUD, DAYD, MOND, YEAD, M_BANK, MEMPAG, MEMBNK, BACPAG, BACBNK
EXTRN  FOR_ADR, REV_ADR, FOR_BNK, REV_BNK, SEACHR, i, j
EXTRN  COM_VOL, TRN_ADR, VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT
EXTRN  CE_YEAD, CE_MOND, CE_DAYD
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DTILSET, DTIPSET, LCD_C, LCD_D, DT_SAV, STA_SEA, S_COMM
EXTRN  SAVADR, DTCOMP, DT_DEL, BANKSET, SEA_FOR, SEA_REV, SEA_FIL, SEA_RIL, ASC_CNV
EXTRN  YEADSP, MONDSP, YDSCNV, MDSCNV, DDSCNV, ASC_DEL
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, MD_CG0F, MD_CG1F, MD_CG2F, MD_CG3F, MD_CG4F, ALM_ONF, IV_CHGF, COMM_F, DS_BRAF
EXTBIT  CR_DSPF, ADENDF, ADSEAF, CMPERF, LCDCHGF

PUBLIC  S_SET

$      INCLUDE (LCDCMD. INC)
$      INCLUDE (LCDT_EX. INC)

$LIST

      DSEG      SADDR
SET_HDT:DS      1                      ;初期設定時データ格納
SET_MDT:DS      1                      ;初期設定分データ格納
SET_SDT:DS      1                      ;初期設定秒データ格納

      BSEG
TIM_OKF DBIT                      ;初期設定時間正常判断フラグ

;*****
;      初期設定処理
;*****
SETSEG      CSEG
S_SET:
      switch(MODE1)
      case 0:
            CALL      !ST_SEL
            break
      case 1:
            CALL      !ST_DAT
            break
      case 2:
            CALL      !ST_TIM
      ends
      RET

;*****
;      初期設定選択処理
;*****
ST_SEL:
      if_bit(MD_CG1F)
            CLR1      MD_CG1F
            HL=#LCDDS                      ;カレンダー・モード選択表示の設定
            DE=#SET_D                      ;LCD 1行目のデータを設定
            CALL      !DTILSET

            DE=#DATE_D                      ;LCD 2行目のデータを設定
            CALL      !DTILSET

            DE=#TIME_D                      ;LCD 3行目のデータを設定
            CALL      !DTILSET

            DE=#CLR1LD                      ;LCD 4行目のデータを設定
            CALL      !DTILSET

            DE=#CLR1LD                      ;LCD 5行目のデータを設定
            CALL      !DTILSET

```

```

DE=#CLR1LD ;LCD 6行目のデータを設定
CALL !DT1LSET

DE=#CLR1LD ;LCD 7行目のデータを設定
CALL !DT1LSET

DE=#CLR1LD ;LCD 8行目のデータを設定
CALL !DT1LSET

MODE2=#0
MODE3=#0
MODE4=#0
SET1 MD_CG2F
SET1 MD_CG3F
SET1 MD_CG4F
CLR1 CR_DSPF
SET1 IV_CHGF ;反転表示行を2行目に設定
SET1 LCDCHGF
INV_DS=#2
endif

if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD) ;キー・コード分岐
case CUR_UP:
if(INV_DS>#2)
INV_DS--
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CUR_DN: ;反転表示行を下に移動
if(INV_DS<#3)
INV_DS++
SET1 IV_CHGF ;反転表示行変更状態に設定
endif
break
case CLEAR: ;前画面モードに変更
MODE0=#0
SET1 MD_CG0F
break
case ENTER: ;反転表示行のモード実行
A=INV_DS
A--
MODE1=A
SET1 MD_CG1F ;モード変更状態に設定
ends
endif
RET

;*****
; 日付設定処理
;*****
ST_DAT:
switch(MODE2)
case 0: ;年設定モード
CALL !ST_DKY0
break
case 1: ;月設定モード
CALL !ST_DKY1
break
case 2: ;日設定モード
CALL !ST_DKY2
ends
RET

;*****
; 年設定処理
;*****
ST_DKY0:
if_bit(MD_CG2F)
CLR1 MD_CG2F
CALL !LCD_DSC ;LCD表示クリア

HL=#LCDDS
DE=#SETDATE ;LCD 1行目のデータを設定

```



```

CALL !DTILSET
HL=#LCDDS+30+2 ;日付設定先頭アドレス
X=YEAD (A) ;年データの取り込み
CALL !YEADSP ;年データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=MOND (A) ;月データの取り込み
CALL !MONDSP ;月データ表示
[HL]=#DIV_SIG (A) ;/
HL++
X=DAYD (A) ;日データの取り込み
CALL !MONDSP ;日データ表示
HL=#LCDDS+60
DE=#YEAR ;LCD 3行目のデータを設定
CALL !DTILSET
CUR_ADR=#93
SETI CR_DSPF
SETI IV_CHGF
SETI LCDCHGF
INV_DS=#0
endif

if_bit(KY_SIGF)
CLR1 KY_SIGF
if(KEY_CD<'0' || KEY_CD>'9') ;数字コード以外?
switch(KEY_CD) ;キーコード分岐
case CLEAR:
MODE1=#0 ;前面モードに変更
SETI MD_CG1F
break
case DELETE:
HL=#LCDDS
B=CUR_ADR (A)
[HL+B]=#SPC (A) ;カーソル・アドレスにスペースを設定
break
case CUR_RI:
if(CUR_ADR<#96) ;年データ4桁目入力
CUR_ADR++ ;カーソル・アドレスに1加算
endif
break
case CUR_LE:
if(CUR_ADR>#93) ;年データ1桁目入力
CUR_ADR-- ;カーソル・アドレスから1減算
endif
break
case ENTER:
MODE2=#1 ;表示年データの登録
SETI MD_CG2F ;モード変更状態に設定
ends
else
B=CUR_ADR (A) ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A) ;数値データを設定
if(CUR_ADR<#96) ;設定データ最下位桁以内
CUR_ADR++
endif
endif
endif
RET

;*****
; 月設定処理
;*****
ST_DKY1:
if_bit(MD_CG2F)
CLR1 MD_CG2F

HL=#LCDDS+120
DE=#MONTH ;LCD 5行目のデータを設定
CALL !DTILSET
SETI LCDCHGF
CUR_ADR=#153
endif

if_bit(KY_SIGF)

```

```

CLR1      KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9')      ;数字コード以外?
switch(KEY_CD)                      ;キー・コード分岐
case CLEAR:                          ;前面モードに変更
    MODE1=#0
    SET1  MD_CG1F
    break
case DELETE:
    HL=#LCDDS
    B=CUR_ADR (A)
    [HL+B]=#SPC (A)                  ;カーソル・アドレスにスペースを設定
    break
case CUR_RI:
    if(CUR_ADR<#154)                 ;月データ2桁目入力
        CUR_ADR++                    ;カーソル・アドレスに1加算
    endif
    break
case CUR_LE:
    if(CUR_ADR>#153)                 ;月データ1桁目入力
        CUR_ADR--                    ;カーソル・アドレスから1減算
    endif
    break
case ENTER:                          ;表示月データの登録
    MODE2=#2
    SET1  MD_CG2F                    ;モード変更状態に設定
ends
else
    B=CUR_ADR (A)                    ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)                ;数値データを設定
    if(CUR_ADR<#154)                 ;設定データ最下位桁以内
        CUR_ADR++
    endif
endif
endif
RET

;*****
;          日設定処理
;*****
ST_DRY2:
    if_bit(MD_CG2F)
        CLR1  MD_CG2F

        HL=#LCDDS+180
        DE=#DAY                       ;LCD7行目のデータを設定
        CALL  IDTILSET
        CUR_ADR=#213
        SET1  LCDCHGF
    endif

IF_BIT(KY_SIGF)
    CLR1  KY_SIGF
    if(KEY_CD<#'0' || KEY_CD>#'9')      ;数字コード以外?
        switch(KEY_CD)                  ;キー・コード分岐
        case CLEAR:                      ;前面モードに変更
            MODE1=#0
            SET1  MD_CG1F
            break
        case DELETE:
            HL=#LCDDS
            B=CUR_ADR (A)
            [HL+B]=#SPC (A)              ;カーソル・アドレスにスペースを設定
            break
        case CUR_RI:
            if(CUR_ADR<#214)              ;日データ2桁目入力
                CUR_ADR++                ;カーソル・アドレスに1加算
            endif
            break
        case CUR_LE:
            if(CUR_ADR>#213)              ;日データ1桁目入力
                CUR_ADR--                ;カーソル・アドレスから1減算
            endif
            break
        case ENTER:

```

```

HL=#LCDDS+93          ;年データの登録
CALL !YDSCNV
YEA=D=A
HL=#LCDDS+153        ;月データの登録
CALL !MNSCNV
MOND=A
HL=#LCDDS+213        ;日データの登録
CALL !DDSCNV
DAYD=A
MODE1=#0
SETI MD_CG1F

ends
else
B=CUR_ADR (A)        ;カーソル・アドレスの設定
HL=#LCDDS
[HL+B]=KEY_CD (A)    ;数値データを設定
if(CUR_ADR<#214)     ;設定データ最下位桁以内
CUR_ADR++
endif
endif
ENDIF
RET

;*****
;          時間設定処理
;*****
ST_TIM:
switch(MODE2)
case 0:              ;時設定モード
CALL !ST_TKY0
break
case 1:              ;分設定モード
CALL !ST_TKY1
break
case 2:              ;秒設定モード
CALL !ST_TKY2
ends
HL=#LCDDS+30+2
A=HOUD+1             ;時(上位)データ設定
CALL !ASC_CNV
A=HOUD               ;時(下位)データ設定
CALL !ASC_CNV
[HL]=#COLON (A)
HL++
A=MIND+1             ;分(上位)データ設定
CALL !ASC_CNV
A=MIND               ;分(下位)データ設定
CALL !ASC_CNV
[HL]=#COLON (A)
HL++
A=SECD+1             ;秒(上位)データ設定
CALL !ASC_CNV
A=SECD               ;秒(下位)データ設定
CALL !ASC_CNV
RET

;*****
;          時設定処理
;*****
ST_TKY0:
if_bit(MD_CG2F)
CLR1 MD_CG2F
CALL !LCD_DSC        ;LCD表示クリア

HL=#LCDDS
DE=#SETTIME         ;LCD1行目のデータを設定
CALL !DTILSET
HL=#LCDDS+60
DE=#HOUR             ;LCD3行目のデータを設定
CALL !DTILSET
CUR_ADR=#93
SETI CR_DSPF
SETI IV_CHGF
INV_DS=#0
SETI LCDCHGF

```

```

endif
if_bit(KY_SIGF)
  CLR1  KY_SIGF
  if(KEY_CD<#'0' || KEY_CD>#'9')
    switch(KEY_CD)
      case CLEAR:
        MODE1=#0
        SET1  MD_CG1F
        break
      case DELETE:
        HL=#LCDDS
        B=CUR_ADR (A)
        [HL+B]=#SPC (A)
        break
      case CUR_RI:
        if(CUR_ADR<#94)
          CUR_ADR++
        endif
        break
      case CUR_LE:
        if(CUR_ADR>#93)
          CUR_ADR--
        endif
        break
      case ENTER:
        HL=#LCDDS+93
        D=#24H
        CALL  !TIM_CHK
        if_bit(TIM_OKF)
          SET_HDT=A

        MODE2=#1
        SET1  MD_CG2F
      endif
    ends
  else
    B=CUR_ADR (A)
    HL=#LCDDS
    [HL+B]=KEY_CD (A)
    if(CUR_ADR<#94)
      CUR_ADR++
    endif
  endif
endif
RET

TIM_CHK:
CLR1  TIM_OKF
if([HL+1]!=#SPC) (A)
  X=A
  A=[HL]
else
  X=[HL] (A)
  A=#0
endif
CALL  !ASC_DEL
if(A<D)
  SET1  TIM_OKF
endif
RET

;*****
;          分設定処理
;*****
ST_TKY1:
  if_bit(MD_CG2F)
    CLR1  MD_CG2F

    HL=#LCDDS+120
    DE=#MINUTE
    ;LCD5行目のデータを設定

```

```

CALL  IDTILSET
SETI  LCDCHGF
CUR_ADR=#153
endif

if_bit(KY_SIGF)
CLR1  KY_SIGF
if(KEY_CD<#'0' || KEY_CD>#'9')      ;数字コード以外?
switch(KEY_CD)                       ;キーコード分岐
case CLEAR:
    MODE1=#0                          ;前面モードに変更
    SETI  MD_CG1F
    break
case DELETE:
    HL=#LCDDS
    B=CUR_ADR (A)
    [HL+B]=#SPC (A)                   ;カーソル・アドレスにスペースを設定
    break
case CUR_RI:
    if(CUR_ADR<#154)                 ;分データ2桁目入力
        CUR_ADR++                   ;カーソル・アドレスに1加算
    endif
    break
case CUR_LE:
    if(CUR_ADR>#153)                 ;分データ1桁目入力
        CUR_ADR--                   ;カーソル・アドレスから1減算
    endif
    break
case ENTER:

    HL=#LCDDS+153
    D=#60H
    CALL  !TIM_CHK
    if_bit(TIM_OKF)
        SET_MDT=A

    MODE2=#2                          ;表示分データの登録
    SETI  MD_CG2F                      ;モード変更状態に設定

endif

ends
else
    B=CUR_ADR (A)                     ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)                 ;数値データを設定
    if(CUR_ADR<#154)                 ;設定データ最下位桁以内
        CUR_ADR++
    endif
endif
endif
endif
RET

;*****
;          秒設定処理
;*****
ST_TKY2:
if_bit(MD_CG2F)
CLR1  MD_CG2F
HL=#LCDDS+180
DE=#SECOND                          ;LCD 7行目のデータを設定
CALL  IDTILSET
SETI  LCDCHGF
CUR_ADR=#213
endif

IF_BIT(KY_SIGF)
CLR1  KY_SIGF
IF(KEY_CD<#'0' || KEY_CD>#'9')      ;数字コード以外?
switch(KEY_CD)                       ;キーコード分岐
case CLEAR:
    MODE1=#0                          ;前面モードに変更
    SETI  MD_CG1F
    break

```

```

case DELETE:
    HL=#LCDDS
    B=CUR_ADR (A)
    [HL+B]-#SPC (A)           ;カーソル・アドレスにスペースを設定
    break
case CUR_RI:
    if(CUR_ADR<#214)         ;秒データ 2 桁目入力
        CUR_ADR++          ;カーソル・アドレスに 1 加算
    endif
    break
case CUR_LE:
    if(CUR_ADR>#213)         ;秒データ 1 桁目入力
        CUR_ADR--          ;カーソル・アドレスから 1 減算
    endif
    break
case ENTER:

    HL=#LCDDS+213
    D=#60H
    CALL 1TIM_CHK
    if_bit(TIM_OKF)
        SET_SDT=A
    MODE1=#0
    SET1 MD_CG1F
    SET1 LCDCHGF
    A=SET_HDT
    A>>=1
    A>>=1
    A>>=1
    A>>=1
    HOUD+1=A                 ;時(上位)に設定
    A=SET_HDT
    A&=#0FH
    HOUD=A                   ;時(下位)に設定
    A=SET_MDT
    A>>=1
    A>>=1
    A>>=1
    A>>=1
    MIND+1=A                 ;分(上位)に設定
    A=SET_MDT
    A&=#0FH
    MIND=A                   ;分(下位)に設定
    A=SET_SDT
    A>>=1
    A>>=1
    A>>=1
    A>>=1
    SECD+1=A                 ;秒(上位)に設定
    A=SET_SDT
    A&=#0FH
    SECD=A                   ;秒(下位)に設定
    endif
ends
ELSE
    B=CUR_ADR (A)           ;カーソル・アドレスの設定
    HL=#LCDDS
    [HL+B]=KEY_CD (A)       ;数値データを設定
    if(CUR_ADR<#214)       ;設定データ最下位桁以内
        CUR_ADR++
    endif
ENDIF
ENDIF
RET
END

```

4.10 通信モード

通信モードでは、名刺（個人）、名刺（会社）、メモ帳で使用するデータの送受信を、PC - 9801シリーズと、または電子手帳同士で行っています。PC - 9801シリーズとの通信はRS - 232Cを介して行い、電子手帳同士の通信はμPD78014の3線式モードを使用して行います。

通信モードへの移行は、COMMキー入力により通信モードになっている間（COMM_Fセット中）行います。送信の場合、データ量は、送信モードに移行するときの状態により異なります（表4 - 12参照）。送信処理を実行するときは、送信開始するデータ登録メモリの先頭アドレスの設定（TRNADR）と、送信するデータ数の設定（TRN_VOL）を行います。受信の場合、受信モードに移行するときの状態に関係なく、すべてのデータを受信します。そして、受信したデータの登録、およびデータ数のカウントをします。また、通信が終了したときは、COMMキー入力前の状態に戻します（COMM_Fクリア）。

表4 - 12 送信時のデータ量の違い

送信モード移行前の状態		送信データ量
メニュー・モード		全登録データ
名刺（個人）、	モード選択	移行しているモードの全データ
名刺（会社）、	検索モードの	表示データ1ページ分
メモ帳モード	データ表示中	

4.10.1 モード選択条件

通信モードの遷移状態の条件を表4 - 13に示します。

表4 - 13 通信モードのTRMOD1-TRMOD3の状態

TRMOD1	TRMOD2	TRMOD3	遷移モード
0	-	-	送受信モード選択
1	-	0	送信先選択モード
	TRN	1	PC - 9801シリーズ送信モード
	RCV		PC - 9801シリーズ受信モード
	TRN	2	電子手帳送信モード
	RCV		電子手帳受信モード
-	3	通信エラー表示モード	

備考 - : don't care

4.10.2 PC - 9801シリーズとのデータ通信

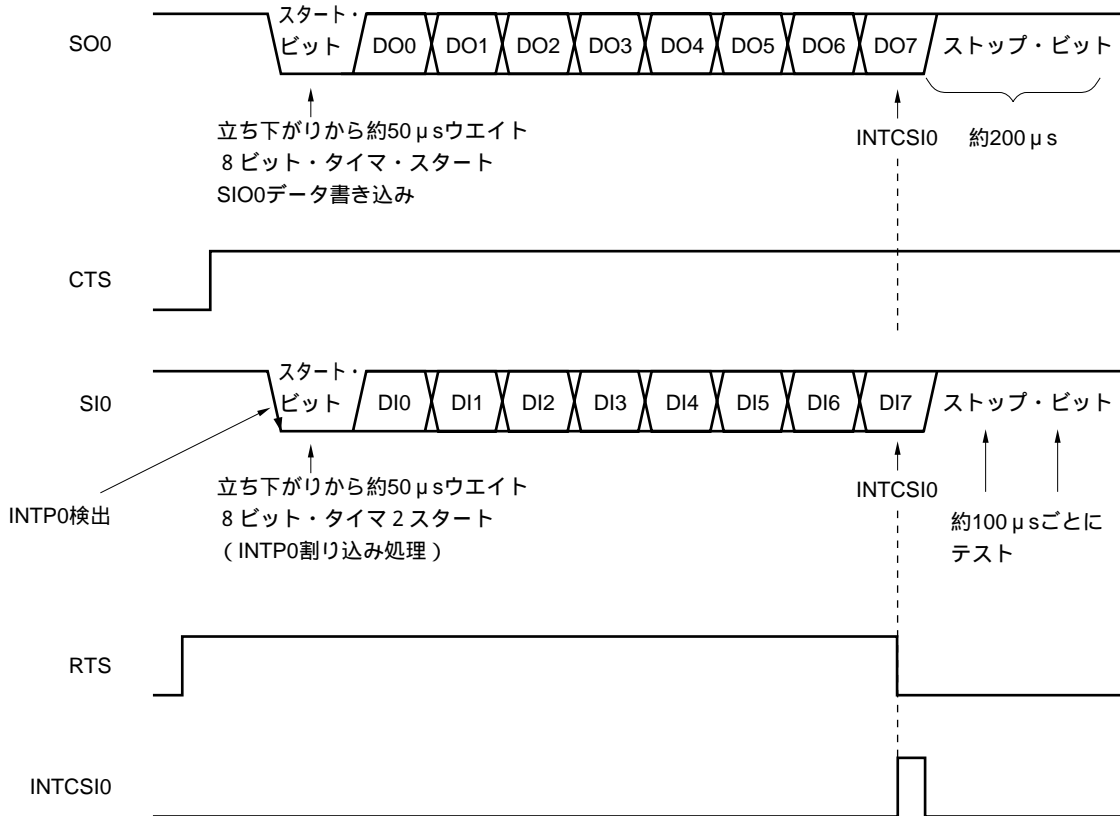
PC - 9801シリーズとのデータ通信では、RS - 232Cインタフェースを使用して行います。データの送受信は、図4 - 20に示すようなフォーマットで行っています。したがって、PC - 9801シリーズ側の通信制御プログラムが必要になります。データの通信フォーマットは、9600 bps、データ8ビット、パリティなし、ストップ・ビット2ビットで行います。通信プログラムの注意点を次に示します。

チェック・サム・データが1バイトのため制御コードが送られてきます。したがって、チェック・サム・データの制御コードを無視する処理が必要になります。

CTS、RTSによるハンドシェイクをしてください。

図4 - 20 PC - 9801シリーズの通信フォーマット

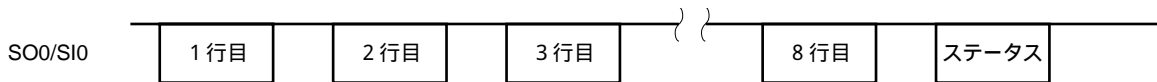
(a) 1バイトのデータ通信



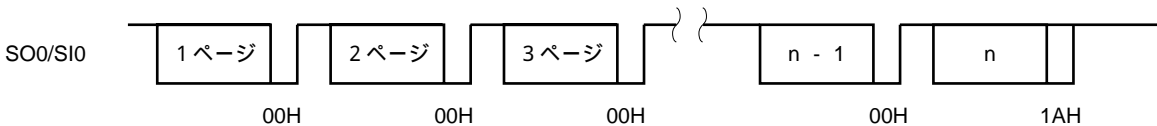
(b) LCD表示1行のデータ通信



(c) LCD表示1ページのデータ通信



(d) LCD表示nページのデータ通信



次に、送信 / 受信の各処理について説明します。

(1) 送信の場合

スタート・ビットをSOラッチの操作で出力し、タイマ2で出力時間を作ります。

スタート・ビットからタイマ2の要求が1回あったときに(半クロック)、SIO0にデータを設定します。

INTCSI0の割り込み処理で、SOラッチをセットし、ストップ・ビットを出力します。そして、タイマ2の割り込み要求4回で2ビット分の時間を作ります。

- の処理をLCD表示1行分(30バイト)送信します。30バイト送信後にチェック・サム・データを送信します。

- の動作を8回(LCD表示8行分)繰り返します。

の処理後、データのパラメータ値16バイトと、そのチェック・サム・データを送信します。最終ページのときは1AHを、最終ページ以外のときは00Hを送信します。

- の処理をデータ登録メモリのTRN_ADRからTRN_VOLの数だけ送信します。

(2) 受信の場合

INTP0の立ち下がリエッジの割り込みで、スタート・ビットを検出します。

スタート・ビットからタイマ2の要求が1回あったときに(半クロック)、SIO0にデータ(0FFH)を設定します。

INTCSI0の割り込み処理で、タイマ2の割り込み要求の2, 4回目にストップ・ビットをテストします。

- の処理をLCD表示1行分(30バイト)受信します。そして、31バイト受信時にチェック・サム・データの判断をします。

- の動作を8回(LCD表示8行分)繰り返します。

の処理後、データのパラメータ値16バイトと、そのチェック・サム・データを受信します。

の次のデータ(1ページ最終データ)が1AHのときは最終ページとなり通信処理を終了します(COMM_Fをクリア)。1AH以外のときは、データ継続中になりますので、からの処理を繰り返します。

の処理後、データ登録メモリにデータを登録します。そして、受信データの判断をし、登録したデータの数を+1します。また、データがオーバーフローしたときはエラー表示モードにします(TRMOD3=3)。

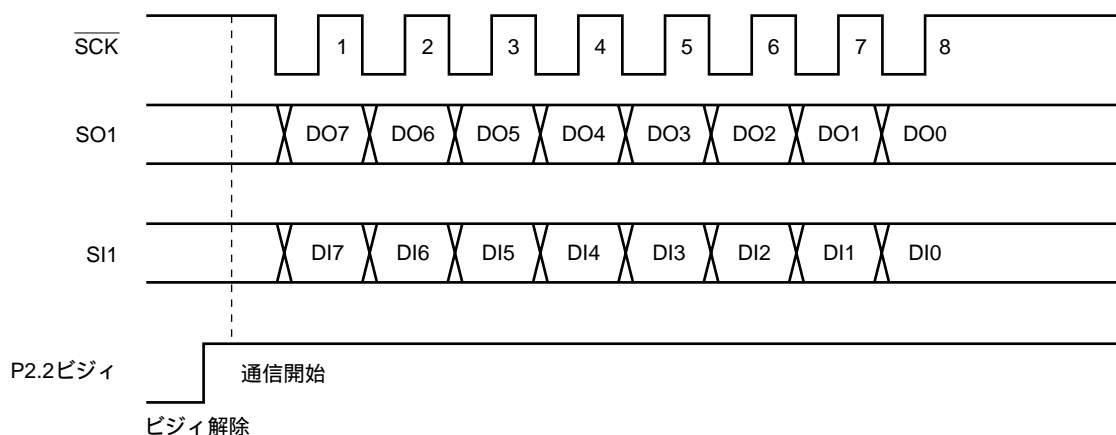
4.10.3 電子手帳同士のデータ通信

電子手帳同士の通信では、3線式のシリアル・インタフェースを使用します。通信方式としては、双方向、半二重で行います。また、シリアル・クロックは、送信側が65.5 kHzを出力します。そして、ハンドシェイク用にビジィ信号の入出力の2線を追加しています。

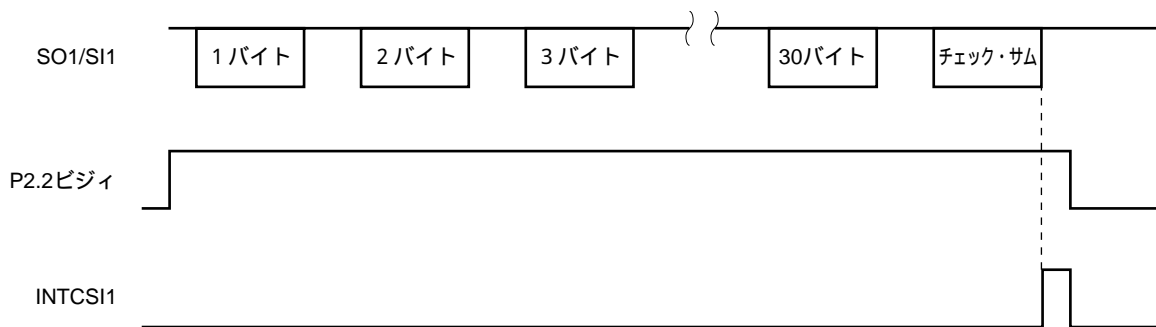
図4-21に通信フォーマットを示します。

図4-21 電子手帳同士の通信フォーマット

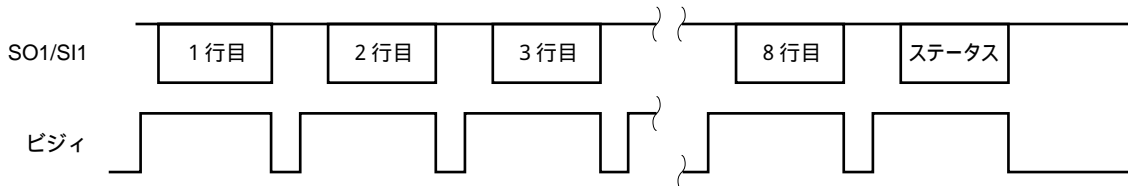
(a) 1バイトのデータ通信



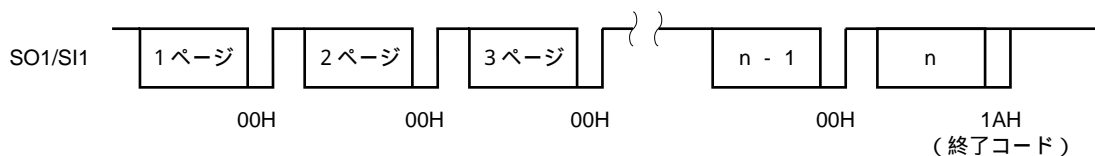
(b) LCD表示1行のデータ通信



(c) LCD表示1ページのデータ通信



(d) LCD表示nページのデータ通信



次に、送信 / 受信の各処理について説明します。

(1) 送信の場合

自動送受信バッファに1行分データ(30バイト)と、チェック・サム・データ1バイトを設定します。

P.2.2によりビジー状態の解除を確認します。

SIO1にデータ (FFH) を書き込み、送信を開始します。

- の動作を8回 (LCD表示8行分) 繰り返します。

の処理後、データのパラメータ値16バイトと、そのチェック・サム・データを設定します。そして、18バイト目には、最終ページのときに1AHを、最終ページ以外ときは00Hを設定して送信します。

- の処理をデータ登録メモリのTRN_ADRからTRN_VOLの数だけ送信します。

(2) 受信の場合

INTCSI1の割り込み処理で、LCD表示1行分 (30バイト) のデータと、チェック・サム・データ1バイトを受信します。受信後にはチェック・サム・データの判断をします。

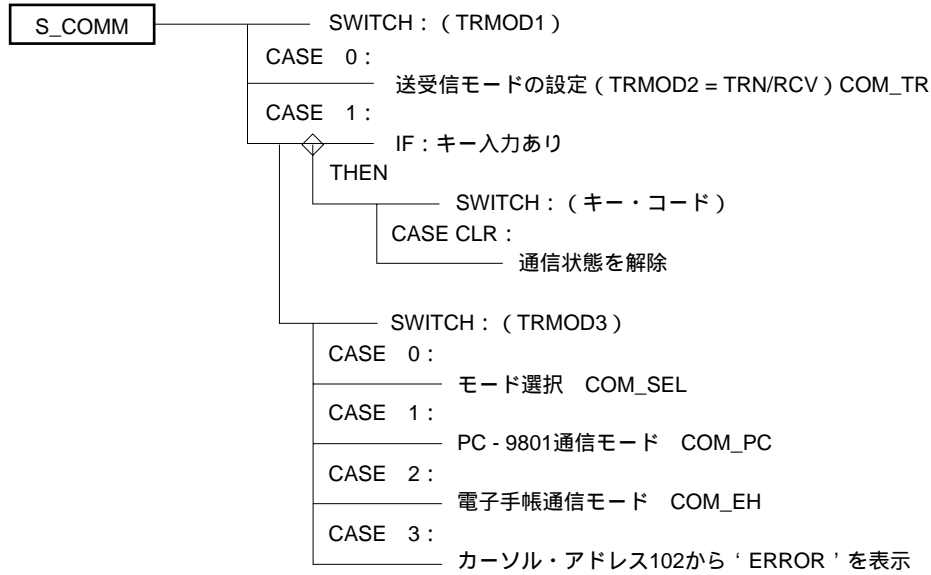
の動作を8回 (LCD表示8行分) 繰り返します。

の処理後、データのパラメータ値16バイトと、そのチェック・サム・データと、通信終了コードのデータを受信します。最後のデータ (1ページ最終データ) が、1AHのときは最終ページとなり通信処理を終了します (COMM_Fをクリア)。1AH以外ときは、データ継続中になりますので、からの処理を繰り返します。

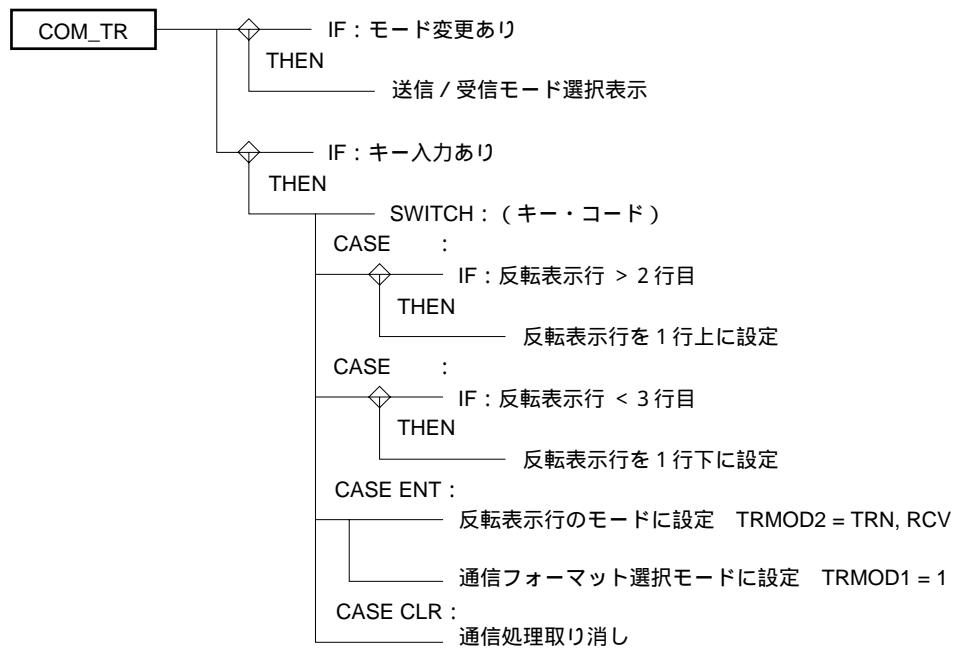
の処理後、データ登録メモリにデータを登録します。そして、受信データの判断をし、登録したデータの数を+1します。また、データがオーバーフローしたときはエラー表示モードにします (TRMOD3 = 3)。

4.10.4 SPDチャート

通信モード

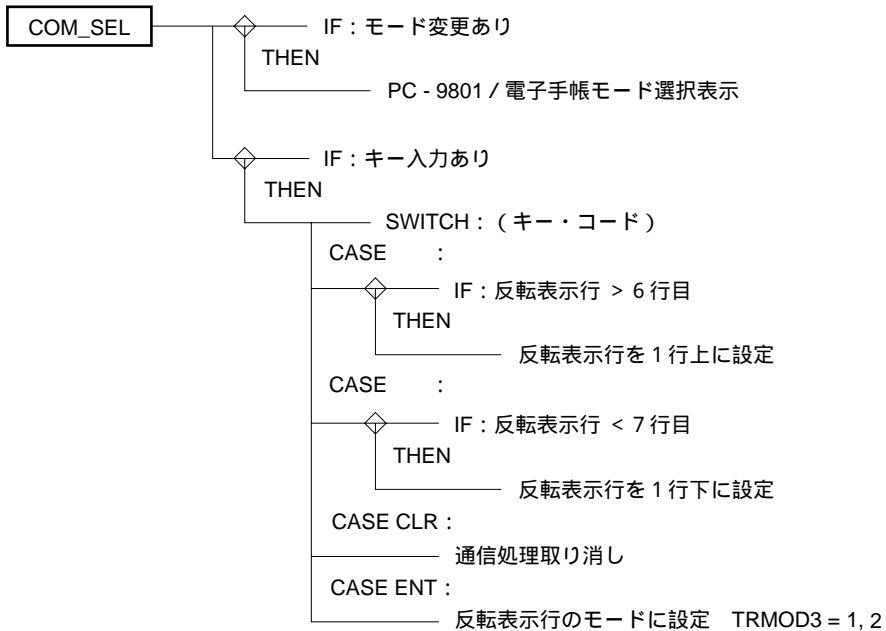


・モード選択



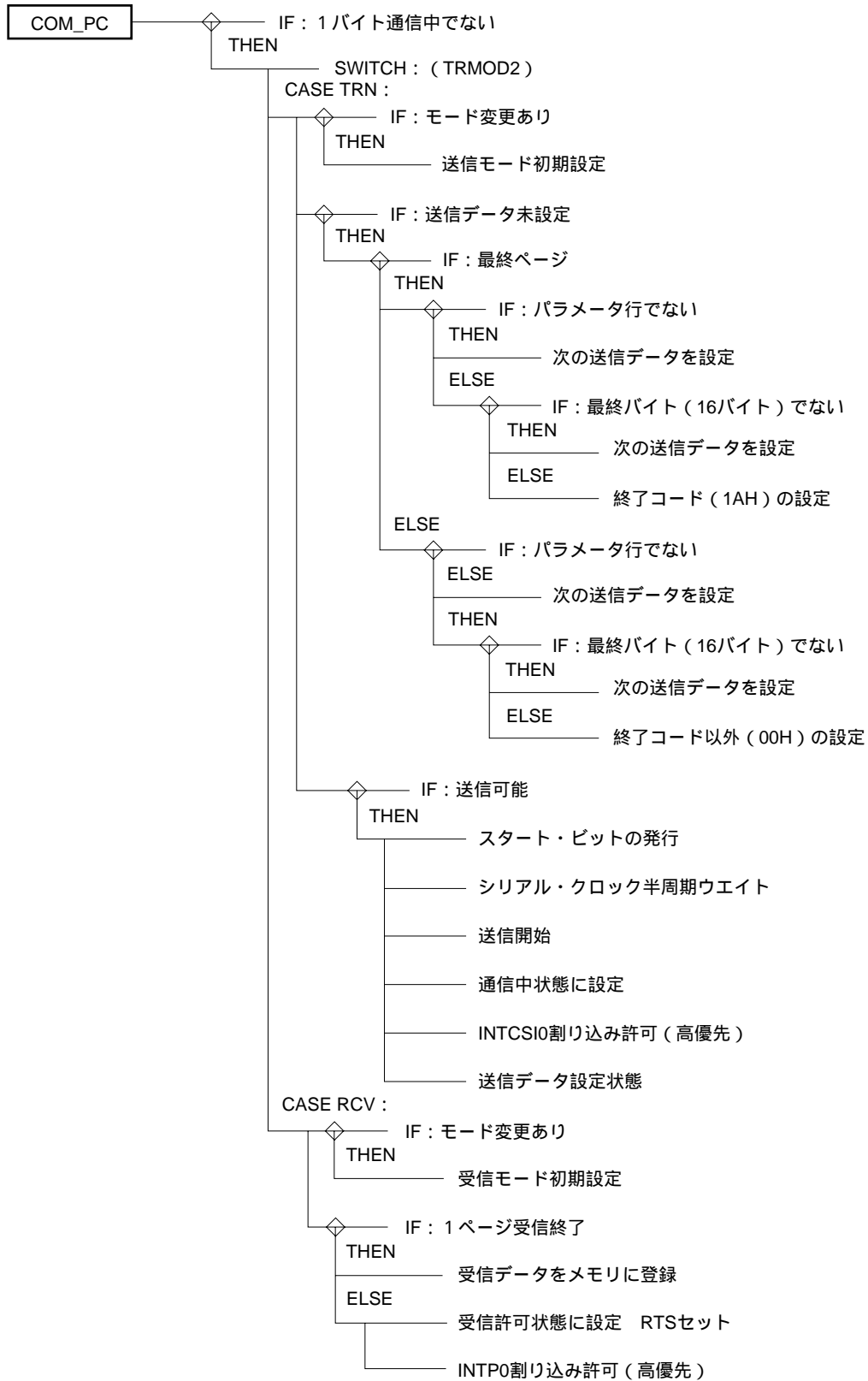
通信モード

・モード選択



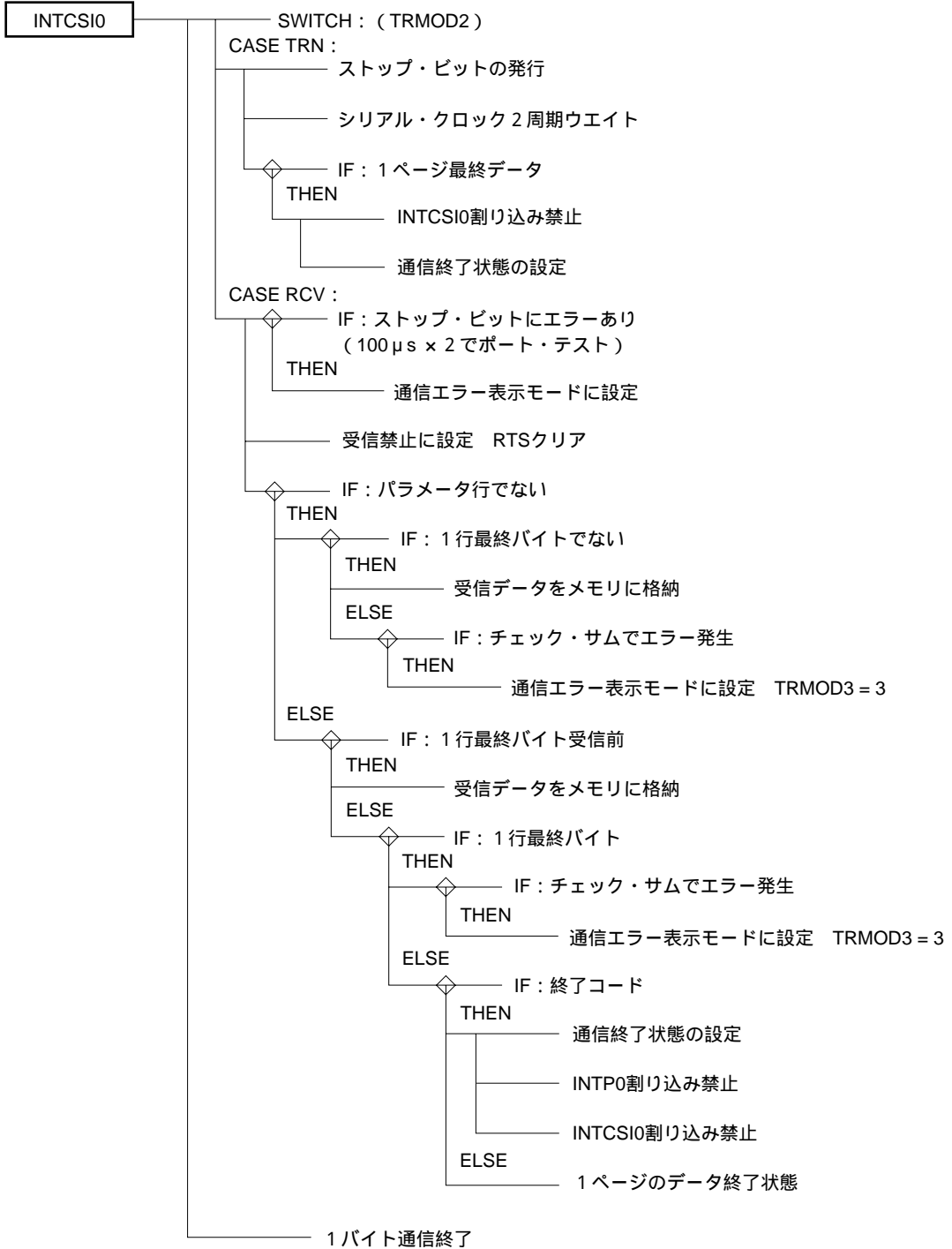
通信モード

・PC - 9801との通信



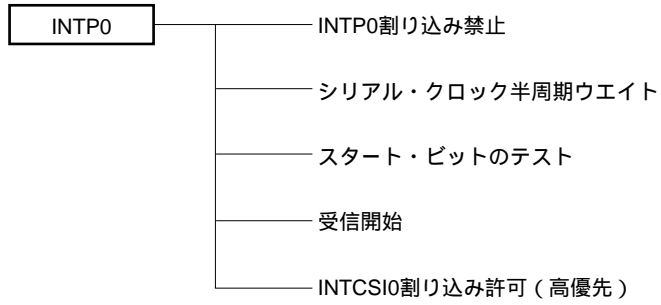
通信モード

・PC-9801との通信



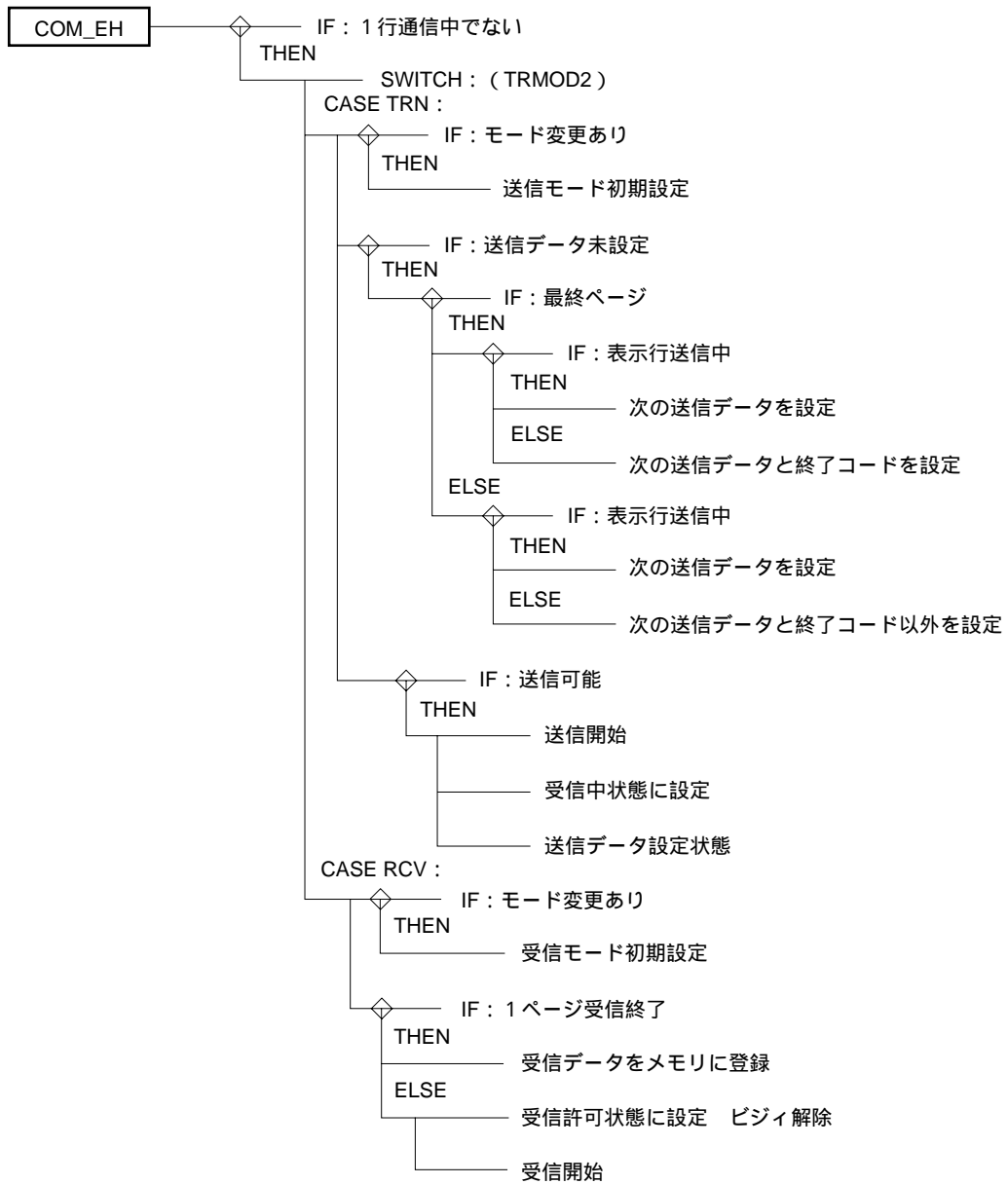
通信モード

・PC - 9801との通信



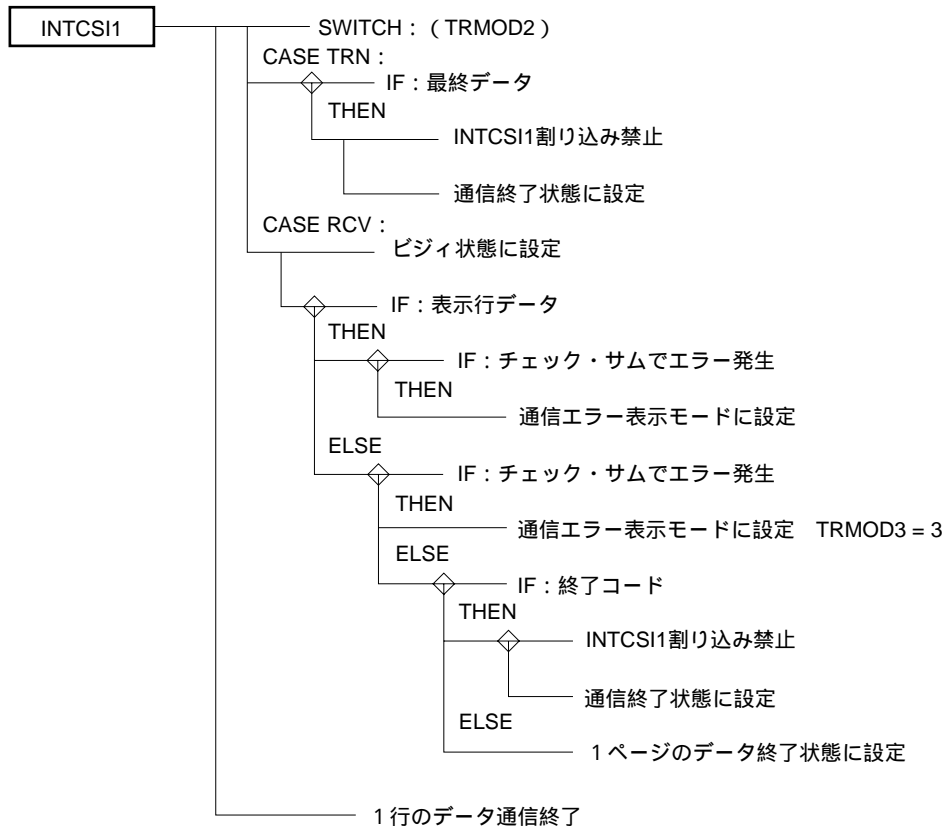
通信モード

・電子手帳同士の通信



通信モード

・電子手帳同士の通信



4.10.5 プログラム・リスト

```

$PC(014)

$NOLIST

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST
EXTRN  S_CURU, S_CURD, S_CURL, S_CURR, S_DEL, S_UP, S_DOWN
EXTRN  DTWORK, LCDDS, LCDCLR, LCD_DSC, DT1LSET, DT1PSET, LCD_C, LCD_D, DT_SAV, STA_SEA
EXTRN  ADM_VOL, ADC_VOL, MEM_VOL

EXTRN  VAC_DAT, ADM_DAT, ADC_DAT, MEM_DAT, SCA_DAT, HOL_DAT
EXTRN  SAVADR, DTCOMP, DT_DEL, DEL_1L, BANKSET, ADR_SEA, AD_SEAR, ADR_INC, ADR_DEC, DT_SET
EXTRN  SEA_FOR, SEA_REV, SEA_FIL, SEA_R1L, M_BANK, MEMPAG, INV_DS, KEY_CD, i
EXTRN  ADRSTA, ADREND

EXTBIT  KY_SIGF, IV_CHGF, ADSEAF, LCDCHGF

PUBLIC  S_COMM, COM_VOL, TRN_ADR, TRN_BNK, COMM_F, TR_CG1F, TR_CG2F, TR_CG3F

$      INCLUDE (LCDCMD. INC)
$      INCLUDE (PORT. INC)
$      INCLUDE (LCDT_EX. INC)

$LIST

;*****
;      ベクタ・テーブル定義
;*****
VEP0   CSEG   AT 06H
      DW      INTP0           ;INTP0 エッジ検出用ベクタ・テーブルの設定
VECS10 CSEG   AT 0EH
      DW      INTCS10        ;シリアル・インタフェース0用ベクタ・テーブルの設定
VECS11 CSEG   AT 10H
      DW      INTCS11        ;シリアル・インタフェース1用ベクタ・テーブルの設定

;*****
;      通信モード選択データ定義
;*****
TRN    EQU    05H           ;送信モード・データ
RCV    EQU    0AH           ;受信モード・データ
EOF    EQU    1AH           ;通信最終コード
CONT   EQU    00H           ;通信継続コード

;*****
;      通信モード使用RAM定義
;*****
      DSEG    SADDR
COM_VOL:DS 1           ;通信データのページ数
TRN_BNK:DS 1           ;通信開始アドレスのメモリ・バンク
TRMOD1: DS 1           ;モード選択 (ネスティング・レベル1)
TRMOD2: DS 1           ;モード選択 (ネスティング・レベル2)
TRMOD3: DS 1           ;モード選択 (ネスティング・レベル3)
TRNDAT: DS 1           ;送信データ格納
LIN_CT: DS 1           ;通信データの表示行数カウンタ
CHR_CT: DS 1           ;通信データの文字数カウンタ
SUM_DT: DS 1           ;チェック・サム・データ
WORKADR:DS 1           ;受信データ格納アドレス

      DSEG    SADDRP
TRNADRP:
TRNADRL:DS 2           ;通信開始アドレス

TRN_ADR EQU    TRNADRL+1

      BSEG
COMM_F DBIT           ;通信処理中フラグ
TR_CG1F DBIT          ;モード1変更フラグ
TR_CG2F DBIT          ;モード2変更フラグ
TR_CG3F DBIT          ;モード3変更フラグ
COMBYTF DBIT          ;1バイト通信中フラグ
COMLINF DBIT          ;1行通信中フラグ
TRNSETF DBIT          ;送信データ設定フラグ
RCVENDF DBIT          ;1ページ受信終了フラグ

```

```

TR_ENDF DBIT ;通信終了フラグ

AT_ADR EQU OFAC0H ;自動転送メモリ・スタート・アドレス

;*****
; 通信処理
;*****
CONSEG CSEG
S_COMM:
    if_bit(TR_CG1F)
        CLR1 TR_CG1F
        HL=#LCDDS ;メニュー表示データの設定
        DE=#COMMUN ;LCD 1行目のデータを設定
        CALL !DTILSET

        DE=#TRANS ;LCD 2行目のデータを設定
        CALL !DTILSET

        DE=#RECEIVE ;LCD 3行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD ;LCD 4行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD ;LCD 5行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD ;LCD 6行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD ;LCD 7行目のデータを設定
        CALL !DTILSET

        DE=#CLR1LD ;LCD 8行目のデータを設定
        CALL !DTILSET

        SET1 TR_CG2F
        SET1 TR_CG3F
        TRMOD1=#0
        TRMOD2=#0
        TRMOD3=#0
        SET1 IV_CHGF ;反転表示行を2行目に設定
        INV_DS=#2
        LIN_CT=#0
        CHR_CT=#0
        CLR1 TR_ENDF
        SET1 LCDCHGF
    endif

    if_bit(TR_ENDF) ;通信終了
        CLR1 TR_ENDF
        CLR1 COMM_F
        SET1 RS_STN ;μPD4711スタンバイに設定
        SET1 CSINK0 ;通信使用制り込み禁止
        SET1 CSINK1
        SET1 PNK0
    endif

    if_bit(KY_SIGF)
        switch(KEY_CD)
            case CLEAR: ;前モードに設定
                CLR1 KY_SIGF
                CLR1 COMM_F
                SET1 RS_STN ;μPD4711スタンバイに設定
                SET1 CSINK0 ;通信使用制り込み禁止
                SET1 CSINK1
                SET1 PNK0
            ends
        endif

    switch(TRMOD1)
        case 0:
            CALL !CON_TR ;送受信モードの設定
            break
        case 1:
            switch(TRMOD3)

```

```

case 0:
    CALL !COM_SEL      ;通信先選択モード
    break
case 1:
    CALL !COM_PC      ;PC-9801通信モード
    break
case 2:
    CALL !COM_BH      ;電子手帳同士の通信モード
    break
case 3:
    if_bit(TR_CG3F)
        SET1 CSIMK0      ;通信使用割り込み禁止
        SET1 CSIMK1
        SET1 RTS          ;受信禁止状態
        SET1 PMK0        ;INTP0割り込み禁止
        CLR1 TR_CG3F
        CALL !LCD_DSC
        !LCDDS+102=#'E' (A) ;通信エラーの表示 'ERROR'
        !LCDDS+103=#'R' (A)
        !LCDDS+104=#'R' (A)
        !LCDDS+105=#'O' (A)
        !LCDDS+106=#'R' (A)
        SET1 LCDCHGF
    endif
ends
ends
RET

;*****
;      転送モードの選択処理
;*****
COM_TR:
    if_bit(KY_SIGF)
        CLR1 KY_SIGF
        switch(KEY_CD)
        case CUR_UP:
            if(INV_DS>#2)
                INV_DS--
                SET1 IV_CHGF      ;反転表示行変更状態に設定
            endif
            break
        case CUR_DN:
            if(INV_DS<#3)
                INV_DS++
                SET1 IV_CHGF      ;反転表示行変更状態に設定
            endif
            break
        case ENTER:
            if(INV_DS==#2)
                TRMOD2=#TRN      ;反転表示行2行目
                ;送信モードに設定
                HL=#LCDDS+60      ;3行目のアドレスを設定
            else
                TRMOD2=#RCV      ;受信モードに設定
                HL=#LCDDS+80      ;2行目のアドレスを設定
            endif
            DE=#CLR1LD          ;設定行の表示をクリア
            CALL !DT1LSET
            SET1 TR_CG2F
            TRMOD1=#1          ;通信処理モードの設定
            TRMOD3=#0
            SET1 TR_CG3F
            SET1 LCDCHGF
        ends
    endif
    RET

;*****
;      転送先選択処理
;*****
COM_SEL:
    if_bit(TR_CG3F)
        CLR1 TR_CG3F
        HL=#LCDDS+150          ;メニュー表示データの設定
        DE=#PC9801            ;LCD8行目のデータを設定
        CALL !DT1LSET

```

```

DE=#ELE_POC ;LCD 7行目のデータを設定
CALL !DTILSET

SETI IV_CHGF ;反転表示行を6行目に設定
INV_DS=#6
SETI LCDCHGF
endif
if_bit(KY_SIGF)
CLR1 KY_SIGF
switch(KEY_CD)
case CUR_UP:
    if(INV_DS>#6)
        INV_DS--
        SETI IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case CUR_DN: ;反転表示行を下に移動:
    if(INV_DS<#7)
        INV_DS++
        SETI IV_CHGF ;反転表示行変更状態に設定
    endif
    break
case ENTER:
    if(INV_DS==#6) ;反転表示行6行目
        TRMOD3=#1 ;PC-9801モードに設定
        SETI RTS ;受信禁止状態
        CLR1 RS_STN ;μPD4711スタンバイ解除
        CLR1 COMBYTF ;通信待ち状態
        HL=#LCDDS+180 ;7行目のアドレスを設定
    else
        TRMOD3=#2 ;電子手帳同士のモードに設定
        CLR1 COMLINF ;通信待ち状態
        HL=#LCDDS+150 ;6行目のアドレスを設定
    endif
    DE=#CLR1LD ;設定行の表示をクリア
    CALL !DTILSET
    CLR1 TR_ENDF ;通信開始
    INV_DS=#0 ;反転表示クリア
    SETI IV_CHGF
ends
endif
RET

;*****
; PC-9801通信処理
;*****
COM_PC:
IF_BIT(!COMBYTF) ;1バイト通信中
switch(TRMOD2) ;通信モードの選択
CASE TRN: ;送信モード
    if_bit(TR.CG2F)
        CLR1 TR.CG2F
        CSIM0=#10000110B ;3線式、LSB、TO2、動作許可
        SETI RELT
        SUM_DT=#0 ;チェック・サム・データをクリア
        CHR_CT=#0
        TRNADR1=#16
        CLR1 TRNSETF ;送信データ未設定に設定
    endif
    IF(COM_VOL!=#0)
        IF_BIT(!TRNSETF) ;送信データ未設定
        IF(COM_VOL==#1) ;最終ページ
            if(LIN_CT<=#7) ;1-7行目転送中
                if(CHR_CT<#30) ;1行データ転送済み
                    CALL !TRNSET0
                    A+=SUM_DT ;チェック・サム・データ作成
                    SUM_DT=A
                    CHR_CT++ ;送信データ数のカウント
                else
                    TRNDAT=SUM_DT(A) ;チェック・サム・データの設定
                    SUM_DT=#0
                    LIN_CT++ ;送信表示行数のカウント
                    CHR_CT=#0
                endif
            endif

```

```

        if(LIN_CT==#8) ;送信表示行 8 行目
            L=#0 ;ステータスの送信
        endif
    else
        if(CHR_CT<#16) ;1 行データ転送済み
            CALL !TRNSET0
            A+=SUM_DT ;チェック・サム・データ作成
            SUM_DT=A
            CHR_CT++ ;送信データ数のカウント
        elseif(CHR_CT==#16)
            TRNDAT=SUM_DT (A) ;チェック・サム・データの設定
            SUM_DT=#0
            CHR_CT++
        else
            TRNDAT=#EOF ;通信最終コードの設定
            CHR_CT++
        endif
    endif
    SETI TRNSETF ;送信データ設定済みに設定
ELSE
    if(LIN_CT<#7) ;1-7 行目転送中
        if(CHR_CT<#30) ;1 行データ転送済み
            CALL !TRNSET0
            A+=SUM_DT ;チェック・サム・データ作成
            SUM_DT=A
            CHR_CT++ ;送信データ数のカウント
        else
            TRNDAT=SUM_DT (A) ;チェック・サム・データの設定
            SUM_DT=#0
            LIN_CT++ ;送信表示行数のカウント
            CHR_CT=#0
        endif
        if(LIN_CT==#8) ;送信表示行 8 行目
            L=#0 ;ステータスの送信
        endif
    else
        if(CHR_CT<#16) ;1 行データ転送済み
            CALL !TRNSET0
            A+=SUM_DT ;チェック・サム・データ作成
            SUM_DT=A
            CHR_CT++ ;送信データ数のカウント
        elseif(CHR_CT==#16)
            TRNDAT=SUM_DT (A) ;チェック・サム・データの設定
            SUM_DT=#0
            CHR_CT++
        else
            TRNDAT=#CONT ;通信継続コードの設定
            M_BANK=TRN_BNK (A) ;送信データのメモリ・バンクを設定
            H=TRN_ADR (A) ;送信データのアドレスを設定
            CALL !ADR_INC ;送信アドレス・インクリメント
            CHR_CT++
            TRN_BNK=M_BANK (A)
            TRNADR=#16
            TRN_ADR=H (A)
        endif
    endif
    SETI TRNSETF ;送信データ設定済みに設定
ENDIF
endif
if_bit(!CTS)
    DI ;スタート・ビット出力処理中割り込み禁止
    SETI CNDT ;スタート・ビット出力
    CLR1 TMIF2
    CLR1 TNC1.1 ;タイマ2 カウント・クリア
    CR20=#108 ;9 6 0 0 b p s に設定
    TOC1=#01100000B ;タイマ2 出力ロウに設定
    SETI TNC1.1 ;タイマ2 動作許可
    CLR1 CSIIF0
    SIO0=TRNDAT (A) ;送信データの設定
    EI
    SETI COMBYTF ;1 バイト送信中に設定
    CLR1 CSINR0 ;シリアル・インタフェース 0 割り込み許可
endif
ELSE
    SETI TR_ENDF

```

```

ENDIF
break
case RCV:
  if_bit(TR_CG2F)
    CLR1 TR_CG2F
    SET1 RELT
    CSINO=#10000110B ;3線式, LSB, TO2, 動作許可
    SUM_DT=#0 ;チェック・サム・データをクリア
    WORKADR=#16
    CLR1 RTS ;受信許可状態
    CLR1 PIFO
    CLR1 PMKO ;INTP0 割り込み許可
  endif
  if_bit(RCVENDF) ;1ページのデータ受信
    CLR1 RCVENDF
    M_BANK=#3
    CALL !BANKSET
    A=!0BF00H
    if(A=#VAC_DAT) ;最終アドレスのメモリ空きあり
      CLR1 ADSEAF
      M_BANK=#0 ;外部メモリを開始位置に設定
      CALL !BANKSET
      H=#ADRSTA
      L=#0
      B=!DTWORK(A) ;比較データを設定
      C=#16 ;1ライン比較の開始アドレスを設定(HL+C)
      DE=#DTWORK+16 ;比較先順アドレス
      CALL !SAVADR ;登録メモリ・アドレスを設定
      DE=#DTWORK
      H=MEMPAG(A)
      L=#0
      switch(!DTWORK)(A) ;受信データの種別
        case ADM_DAT: ;名刺(個人)データ
          ADM_VOL++
          break
        case ADC_DAT: ;名刺(会社)データ
          ADC_VOL++
          break
        case MEM_DAT: ;メモ帳データ
          MEM_VOL++
        ends
      CALL !DT_SAV ;データ登録
      CLR1 PIFO
      CLR1 PMKO ;INTP0 割り込み許可
      CLR1 RTS ;受信許可状態
    else
      SET1 TR_CG3F ;メモリ・アドレスに空きなし
      TRMOD3=#3 ;オーバフロー・モードに設定
    endif
  endif
ends
ENDIF
RET
TRNSET0:
M_BANK=TRN_BNK(A) ;送信データのメモリ・バンクを設定
CALL !BANKSET
HL=TRNADRP(AX) ;送信データのアドレスを設定
TRNDAT=[HL](A) ;送信データの設定
TRNADRL++ ;送信アドレスの更新
RET

;*****
; PC-9801通信シリアル割り込み
;*****
INTCS10:
SEL RB1
switch(TRMOD2)
case TRN:
  CLR1 TMIF2
  while_bit(!TMIF2) ;50μsウエイト
  endw
  SET1 RELT ;ストップ・ビット出力
  CLR1 TMIF2
  for(A=#0;A<#4;A++) ;200μsウエイト
    while_bit(!TMIF2) ;50μs×4
  endf

```



```

        endw
        CLR1    TMIF2
    next
    CLR1    TRNSETF                ;送信データ未設定に設定
    if(LIN_CT==#8)                ;1ページ最終データ
        if(CHR_CT==#18)
            if(TRNDAT==#EOF)      ;データ終了コード
                SET1    TR_ENDF    ;送信終了状態に設定
            endif
            COM_VOL--
            LIN_CT=#0
            CHR_CT=#0
        endif
    endif
    break
CASE RCV:
    SET1    RTS
    CLR1    TMIF2                ;ストップ・ビットのテスト100μs
    while_bit(!TMIF2)           ;50μsウエイト
    endw
    CLR1    TMIF2
    while_bit(!TMIF2)           ;50μsウエイト
    endw
    CLR1    TMIF2
    if_bit(IRS_STA)             ;ストップ・ビットのテスト
    TRMOD3=#3                   ;ストップ・ビットのエラー発生
    SET1    TR_CG3F
    endif

    if(LIN_CT<=#7)              ;1-7行目転送中
        if(CHR_CT<#30)          ;1行データ転送済み
            HL=#DTWORK
            B=WORKADR (A)
            [HL+B]=S100 (A)      ;受信データの格納
            A+=SUM_DT            ;チェック・サム・データ作成
            SUM_DT=A
            WORKADR++
            CHR_CT++            ;受信データ数のカウント
        else
            A=S100               ;チェック・サム・データの取り込み
            if(A!=SUM_DT)        ;チェック・サム・データの比較
                TRMOD3=#3       ;チェック・サム・データのエラー
                SET1    TR_CG3F
            endif
            SUM_DT=#0
            LIN_CT++             ;受信表示行数のカウント
            CHR_CT=#0
            if(LIN_CT==#8)       ;受信表示行8行目
                WORKADR=#0
            endif
        endif
    endif
    if(TRMOD3!=#3)
        CLR1    PIF0
        CLR1    PNK0             ;INTP0 割り込み許可
        CLR1    RTS             ;受信許可状態
    endif
else
    if(CHR_CT<#16)
        HL=#DTWORK
        B=WORKADR (A)
        [HL+B]=S100 (A)        ;受信データの格納
        A+=SUM_DT              ;チェック・サム・データ作成
        SUM_DT=A
        WORKADR++
        CHR_CT++               ;受信データ数のカウント
        if(TRMOD3!=#3)
            CLR1    PIF0
            CLR1    PNK0         ;INTP0 割り込み許可
            CLR1    RTS         ;受信許可状態
        endif
    elseif(CHR_CT==#16)
        A=S100                  ;チェック・サム・データの取り込み
        if(A!=SUM_DT)          ;チェック・サム・データの比較
            TRMOD3=#3          ;チェック・サム・データのエラー
            SET1    TR_CG3F
        endif
    endif

```

```

        SET1      TR.CG3F
    endif
    SUM_DT=#0
    CHR_CT++
    if(TRMOD3!=#3)
        CLR1     PIF0
        CLR1     PMK0      ;INTP0 割り込み許可
        CLR1     RTS      ;受信許可状態
    endif
else
    A=S100      ;1 ページ最終データの取り込み
    if(A=#BOF) ;終了コード
        SET1     TR.ENDF ;通信終了状態に設定
    endif
    SUM_DT=#0
    SET1     RCVENDF      ;1 ページ転送終了状態に設定
    LIM_CT=#0
    CHR_CT=#0
    WORKADR=#16
endif
endif
ends
CLR1     COMBYTF      ;1 バイト通信終了
RETI

;*****
;   PC-9801 通信エッジ割り込み
;*****
INTP0:
    SET1     PMK0
    CLR1     TMIF2
    CLR1     TMC1.1      ;タイマ2カウント・クリア
    CR20=#108           ;9600bpsに設定
    TOC1=#01100000B     ;タイマ2出力ロウに設定
    SET1     TMC1.1      ;タイマ2動作許可
    while_bit(ITMIF2)    ;半クロック(50μs) ウェイト
    endw
    if_bit(IRS_STA)      ;スタート・ビットのテスト
        CLR1     CSIF0
        SIO0=#OFFH      ;受信開始
        SET1     COMBYTF ;1 バイト通信中に設定
        CLR1     CSINK0  ;シリアル・インタフェース0 割り込み許可
    else
        CLR1     PMK0      ;INTP0 割り込み許可
    endif
    RETI

;*****
;   電子手帳同士通信処理
;*****
COM_EH:
    IF_BIT(!COMLINF)    ;1 行通信中
        switch(TRMOD2) ;通信モードの選択
            CASE TRN:   ;通信モード
                if_bit(TR.CG2F)
                    CLR1     TR.CG2F
                    CSIM1=#10100011B ;LSB, 自動送受信許可, 動作許可
                    ADTC=#00000000B ;受信禁止, 単発, エラー/ストロブ/ビジー制御なし
                    TCL3=#1011011B  ;SCK1=65.5kHz 選択
                    SUM_DT=#0
                    TRNADR=#16
                    CLR1     TRNSETF ;送信データ未設定に設定
                    CLR1     COMLINF ;1 行通信終了
                    CLR1     CSIF1
                    CLR1     CSINK1  ;シリアル・インタフェース1 割り込み許可
                endif
                IF(COM_VOL!=#0)
                    IF_BIT(!TRNSETF) ;送信データ未設定
                        IF(COM_VOL==#1) ;最終ページ
                            if(LIN_CT<=#7) ;1-7行目転送中
                                DE=#AT_ADR+30
                                HL=TRNADRP (AX);送信データのアドレスを設定
                                for(i=#0;i<#30;i++)
                                    CALL     ITRNSET1 ;送信データの設定
                                next
                            endif
                        endif
                    endif
                endif
            endc
        endc
    endc

```

```

        TRNADRP=HL (AX) ;送信データのアドレスを格納
        SUM_DT=#0
        ADTP=#31-1
    else
        DE=#AT_ADR+17
        H=TRN_ADR (A) ;送信データのアドレスを格納
        L=#00H
        for(i=#0;i<#16;i++)
            CALL !TRNSET1 ;送信データの設定
        next
        [DE]=SUM_DT (A)
        SUM_DT=#0 ;チェック・サム・データをクリア
        E--
        [DE]=#EOF (A) ;通信最終コードの設定
        ADTP=#18-1
    endif
    SET1 TRNSETF ;送信データ設定済みに設定
ELSE
    if(LIN_CT<=#7) ;1-8行目転送中
        DE=#AT_ADR+30
        HL=TRNADRP (AX) ;送信データのアドレスを設定
        for(i=#0;i<#30;i++)
            CALL !TRNSET1 ;送信データの設定
        next
        [DE]=SUM_DT (A)
        TRNADRP=HL (AX) ;送信データのアドレスを格納
        SUM_DT=#0 ;チェック・サム・データをクリア
        ADTP=#31-1
    else
        DE=#AT_ADR+17
        H=TRN_ADR (A) ;送信データのアドレスを設定
        L=#00H
        for(i=#0;i<#16;i++)
            CALL !TRNSET1 ;送信データの設定
        next
        [DE]=SUM_DT (A)
        SUM_DT=#0 ;チェック・サム・データをクリア
        E--
        [DE]=#CONT (A) ;通信継続コードの設定
        ADTP=#18-1
        CALL !ADR_INC ;送信アドレス・インクリメント
        CALL !BANKSET
        A=H
        X=#16
        TRNADRP=AX ;送信データのアドレスを格納
    endif
    SET1 TRNSETF ;送信データ設定済みに設定
ENDIF
if_bit(BUSYIN)
    SIO1=#0FFH ;転送開始
    SET1 COMLINF ;1行送信中に設定
endif
elseif_bit(!COMLINF)
    if_bit(BUSYIN)
        SIO1=#0FFH ;転送開始
        SET1 COMLINF ;1行送信中に設定
    endif
endif
ELSE
    SET1 TR_ENDF ;通信終了状態に設定
ENDIF
break
CASE RCY:
    if_bit(TR.CG2F)
        CLR1 TR.CG2F
        CSIMi=#10100000B ;LSB, 自動送受信許可, 動作許可, 外部クロック
        ADTC=#10000000B ;受信許可, 単発, エラー/ストロブ/ビジー制御なし
        SUM_DT=#0 ;チェック・サム・データをクリア
        WORKADR=#16
        CLR1 CSIIF1
        CLR1 CSINX1 ;シリアル・インタフェース1割り込み許可
        CLR1 COMLINF
    endif
    if_bit(RCVENDF) ;1ページのデータ受信
        CLR1 RCVENDF

```

```

M_BANK=#3
CALL !BANKSET
A=!0BF00H
if(A=#VAC_DAT) ;最終アドレスのメモリ空きあり
  CLR! ADSEAF
  M_BANK=#0 ;外部メモリを開始位置に設定
  CALL !BANKSET
  H=#ADRSTA
  L=#0
  B=!DTWORK (A) ;比較データを設定
  C=#16 ;1ライン比較の開始アドレスを設定 (HL+C)
  DE=#DTWORK+16 ;比較先頭アドレス
  CALL !SAVADR ;登録メモリ・アドレスを設定
  DE=#DTWORK
  H=#MENPAG (A)
  L=#0
  switch(!DTWORK) (A) ;受信データの種類
  case ADM_DAT: ;名刺(個人)データ
    ADM_VOL++
    break
  case ADC_DAT: ;名刺(会社)データ
    ADC_VOL++
    break
  case MEM_DAT: ;メモ帳データ
    MEM_VOL++
  ends
  CALL !DT_SAV ;データ登録
else
  SET! TR.CG3F ;メモリ・アドレスに空きなし
  TRMOD3=#3 ;オーバーフロー・モードに設定
endif
elseif_bit(!COMLINF)
  if(LIN_CT<=#7)
    ADTP=#31-1
  else
    ADTP=#18-1
  endif
  SIO1=#OFFH
  SET! BUSYOUT ;受信許可状態
  SET! COMLINF ;1行データ通信中
endif
ends
endif
RET

TRNSET1:
M_BANK=TRN_BNK (A) ;送信データのメモリ・バンクを設定
CALL !BANKSET
[DE]=[HL] (A) ;送信データの設定
L++
E-- ;送信アドレスの更新
A+=SUM_DT ;チェック・サム・データ作成
SUM_DT=A
RET

;*****
; 電子手帳同士の通信シリアル割り込み
;*****
INTCS11:
SEL RB1
switch(TRNMOD2)
case TRN:
  CLR! TRNSETF ;送信データ未設定に設定
  if(LIN_CT=#8) ;ステータス・データ行
    if(!IAT_ADR==#EOF) (A) ;データ終了コード
      SET! TR_ENDF ;通信終了状態に設定
    endif
    CHR_CT=#0
    COM_VOL--
    LIN_CT=#0H
  else
    LIN_CT++ ;送信表示行数のカウント
  endif
  break
CASE RCV:

```

```

CLR1  BUSYOUT          ;ビジー状態に設定
if(LIN_CT<=#7)        ;表示行データ
  HL=#AT_ADR+30      ;文字数 = 30文字
  A=#0
  for(i=#0;i<#30;i++) ;チェック・サム・データの作成
    A+=[HL]
    HL--
  next
  if(A==!AT_ADR)      ;チェック・サム・データの比較
    X=LIN_CT(A)      ;行番号より格納アドレスの設定
    A=#30
    AX*=X
    AX+=#DTWORK+16
    HL=AX
    DE=#AT_ADR+30    ;文字数 = 16文字
    for(i=#0;i<#30;i++) ;チェック・サム・データの作成
      [HL]=[DE](A)
      HL++
      DE--
    next
  else
    TRMOD3=#3        ;チェック・サム・データのエラー
    SET1  TR_CG3F
  endif
  LIN_CT++          ;送信表示行数のカウント
else
  HL=#AT_ADR+17      ;文字数 = 16文字
  A=#0
  for(i=#0;i<#16;i++) ;チェック・サム・データの作成
    A+=[HL]
    HL--
  next
  if(A==!AT_ADR+1)   ;チェック・サム・データの比較
    DE=#AT_ADR+17    ;文字数 = 16文字
    HL=#DTWORK
    for(i=#0;i<#16;i++) ;チェック・サム・データの作成
      [HL]=[DE](A)
      HL++
      DE--
    next
    SET1  RCVENDF      ;1 ページ転送終了状態に設定
    if(!AT_ADR==#EOF)(A) ;終了コード
      SET1  CSIMK1     ;シリアル1 割り込み禁止
      SET1  TR_ENDF    ;通信終了状態に設定
    endif
  else
    TRMOD3=#3        ;チェック・サム・データのエラー
    SET1  TR_CG3F
  endif
  LIN_CT=#0H
endif
ends
CLR1  COMLINF        ;1 行通信終了
RETI
END

```

4.11 表示データ・テーブル

表示データ・テーブルでは、各動作でのモード選択時における表示データを、ROMに用意しています。データ形式としては、LCD表示1行分のデータを単位とした、30文字で構成しています。データ形式を図4 - 22に示します。

図4 - 22 LCD表示データ形式



4.11.1 プログラム・リスト

```

$PC(014)

EXTRN  CUR_UP, CUR_DN, CUR_LE, CUR_RI, ENTER, DELETE, CLEAR, UP, DOWN
EXTRN  MENU, COMM, OFF, MR, MC, M_ADD, M_SUB, SIG_INV, SPC, PAR_LE, PAR_RI
EXTRN  MUL_SIG, ADD_SIG, SUB_SIG, POINT, DIV_SIG, COLON, EQU_SIG, QUEST

;*****
;   名刺（個人）LCD表示データ
;*****
PUBLIC  ADM_D, REG_D, COR_D, SEA_D, DEL_D, CLRILD, NAME_D, TEL_D, FAX_D, ADR_D
PUBLIC  COMP_D, MEMO_D, NA_SEA, ATOM_D, NTOZ_D, NEXT_P, NAM_DAT, MEM_OV

;*****
;   名刺（会社）LCD表示データ
;*****
PUBLIC  ADC_D, COM_D

;*****
;   カレンダーLCD表示データ
;*****
PUBLIC  CALEND, TOD_D, MONTH_D, SETHOL1, WEEK_D, OPEN_D, END_D, ALARM_D, MEMO_D1
PUBLIC  BUSI_D, YEAR_D1, YEAR_D2, MON_D1, MON_D2, DAY_D, SETHOL2, WEKHOL1
PUBLIC  SETDAT1, WEKHOL2, HOLMAK, WEKMAK, SETDAT2, DELKEY

;*****
;   メモ帳LCD表示データ
;*****
PUBLIC  MEMODSP, TIT_L1, REG_L2, TIT_SEA, T_DAT

;*****
;   世界時計LCD表示データ
;*****
PUBLIC  CLOCK, DISPLAY, CIT_SBL, CITY_D, NEWYORK, LOSANG, BRASIL, LIMA, LONDON
PUBLIC  ROME, CAIRO, MOSCOW, HONOLU, SYDNEY, PEKING, HONG, NEWDEL, KUWAIT

;*****
;   初期設定LCD表示データ
;*****
PUBLIC  SET_D, DATE_D, TIME_D, SETDATE, YEAR, MONTH, DAY, SETTIME, HOUR, MINUTE, SECOND

;*****
;   通信処理表示データ
;*****
PUBLIC  COMMUN, TRANS, RECEIVE, PC9801, BLE_POC

LCDSEG  CSEG
;*****
;   名刺（個人）LCD表示データ
;*****
ADM_D:
  DB    'A', 'D', 'D', 'R', 'E', 'S', 'S', PAR_LE, 'M', 'A'
  DB    'N', PAR_RI, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

REG_D:
  DB    SPC, SPC, 'R', 'E', 'G', 'I', 'S', 'T', 'E', 'R'
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COR_D:
  DB    SPC, SPC, 'C', 'O', 'R', 'R', 'E', 'C', 'T', 'I'
  DB    'O', 'N', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SEA_D:
  DB    SPC, SPC, 'S', 'E', 'A', 'R', 'C', 'H', SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DEL_D:
  DB    SPC, SPC, 'D', 'E', 'L', 'E', 'T', 'E', SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

CLRILD:
  DB    SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

```



```

DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NAME_D:
DB      'N', 'A', 'M', 'E', QUEST, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TEL_D:
DB      'T', 'E', 'L', QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

FAX_D:
DB      'F', 'A', 'X', QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

ADR_D:
DB      'A', 'D', 'D', 'R', 'E', 'S', 'S', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COMP_D:
DB      'C', 'O', 'M', 'P', 'A', 'N', 'Y', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MEMO_D:
DB      'M', 'E', 'M', 'O', QUEST, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NA_SEA:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'N', 'A', 'M', 'E', SPC, 'S', 'E', 'A', 'R', 'C'
DB      'H', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

ATOM_D:
DB      SPC, SPC, 'A', SPC, 'B', SPC, 'C', SPC, 'D', SPC
DB      'E', SPC, 'F', SPC, 'G', SPC, 'H', SPC, 'I', SPC
DB      'J', SPC, 'K', SPC, 'L', SPC, 'M', SPC, SPC, SPC

NTOZ_D:
DB      SPC, SPC, 'N', SPC, 'O', SPC, 'P', SPC, 'Q', SPC
DB      'R', SPC, 'S', SPC, 'T', SPC, 'U', SPC, 'V', SPC
DB      'W', SPC, 'X', SPC, 'Y', SPC, 'Z', SPC, SPC, SPC

NEXT_P:
DB      SPC, SPC, 'N', 'E', 'X', 'T', SPC, 'P', 'A', 'G'
DB      'E', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NAM_DAT:
DB      'N', 'A', 'M', 'E', SPC, 'D', 'A', 'T', 'A', SPC
DB      PAR_LE, SPC, PAR_R1, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MEM_OV:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, 'M'
DB      'E', 'M', 'O', 'R', 'Y', SPC, 'O', 'V', 'E', 'R'
DB      'F', 'L', 'O', 'W', SPC, SPC, SPC, SPC, SPC, SPC

;*****
;   名刺(会社)LCD表示データ
;*****
ADC_D:
DB      'A', 'D', 'D', 'R', 'E', 'S', 'S', PAR_LE, 'C', 'O'
DB      'M', 'P', 'A', 'N', 'Y', PAR_R1, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

COM_D:
DB      'C', 'O', 'M', 'P', 'A', 'N', 'Y', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
    
```



```

*****
; カレンダーLCD表示データ
*****
CALEND:
DB 'C','A','L','E','N','D','A','R',SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

TOD_D:
DB SPC,SPC,'T','O','D','A','Y',SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MONTH_D:
DB SPC,SPC,'M','O','N','T','H',SPC,'D','I'
DB 'S','P','L','A','Y',SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

SETHOL1:
DB SPC,SPC,'S','E','T','H','O','L','I'
DB 'D','A','Y',SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

WEEK_D:
DB SPC,'S','U','N',SPC,'M','O','N',SPC,'T'
DB 'U','E',SPC,'W','E','D',SPC,'T','H','U'
DB SPC,'F','R','I',SPC,'S','A','T',SPC,SPC

OPEN_D:
DB SPC,SPC,'O','P','E','N',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

END_D:
DB SPC,SPC,'E','N','D',QUEST,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

ALARM_D:
DB SPC,SPC,'A','L','A','R','M',QUEST,SPC,SPC
DB SPC,SPC,'O','N',SPC,DIV_SIG,SPC,'O','F','F'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

BUSI_D:
DB SPC,SPC,'B','U','S','I','N','E','S','S'
DB QUEST,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MEMO_D1:
DB SPC,SPC,'M','E','M','O',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

YEAR_D1:
DB 'Y','E','A','R',QUEST,PAR_LE,'1','0','0'
DB '1',SUB_SIG,'2','0','0','0',PAR_RI,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

YEAR_D2:
DB SPC,SPC,'Y','E','A','R',QUEST,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MON_D1:
DB 'M','O','N','T','H',QUEST,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MON_D2:
DB SPC,SPC,'M','O','N','T','H',QUEST,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DAY_D:

```



```

DB      SPC, SPC, 'D', 'A', 'Y', QUEST, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETHOL2:
DB      'S', 'E', 'T', SPC, 'H', 'O', 'L', 'I', 'D', 'A'
DB      'Y', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

WEKHOL1:
DB      SPC, SPC, 'W', 'E', 'E', 'K', 'L', 'Y', SPC, 'H'
DB      'O', 'L', 'I', 'D', 'A', 'Y', SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETDAT1:
DB      SPC, SPC, 'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E'
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

WEKHOL2:
DB      'W', 'E', 'E', 'K', 'L', 'Y', SPC, 'H', 'O', 'L'
DB      'I', 'D', 'A', 'Y', SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

HOLNAK:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'o', COLON, 'H', 'O', 'L', 'I', 'D', 'A', 'Y', SPC

WEKNAK:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      'x', COLON, 'W', 'E', 'E', 'K', 'L', 'Y', SPC

SETDAT2:
DB      'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DELKEY:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, 'D', 'E', 'L', SPC
DB      'K', 'E', 'Y', EQU_SIG, 'C', 'L', 'E', 'A', 'R'

;*****
;      メモ帳LCD表示データ
;*****
MEMODSP:
DB      'M', 'E', 'M', 'O', SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIT_L1:
DB      'T', 'I', 'T', 'L', 'E', PAR_LE, 'L', 'E', 'V', 'E'
DB      'L', 'I', PAR_RI, QUEST, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

REG_L2:
DB      'R', 'E', 'G', 'I', 'S', 'T', 'E', 'R', PAR_LE, 'L'
DB      'E', 'V', 'E', 'L', '2', PAR_RI, QUEST, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIT_SEA:
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, 'T'
DB      'I', 'T', 'L', 'E', SPC, 'S', 'E', 'A', 'R', 'C'
DB      'H', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

T_DAT:
DB      'T', 'I', 'T', 'L', 'E', SPC, 'D', 'A', 'T', 'A'
DB      PAR_LE, SPC, PAR_RI, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

;*****
;      世界時計LCD表示データ
;*****

```

```

CLOCK:
  DB 'C','L','O','C','K',SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

DISPLAY:
  DB SPC,SPC,'D','I','S','P','L','A','Y',SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CIT_SEL:
  DB SPC,SPC,'C','I','T','Y',SPC,'S','E','L'
  DB 'E','C','I',SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CITY_D:
  DB 'C','I','T','Y',SPC,'S','E','L','E','C'
  DB 'T',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

NEWYORK:
  DB SPC,SPC,'N','E','W',SPC,'Y','O','R','K'
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LOSANG:
  DB SPC,SPC,'L','O','S',SPC,'A','N','G','E'
  DB 'L','E','S',SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

BRASIL:
  DB SPC,SPC,'B','R','A','S','I','L','A'
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LIMA:
  DB SPC,SPC,'L','I','M','A',SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

LONDON:
  DB SPC,SPC,'L','O','N','D','O','N',SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

ROME:
  DB SPC,SPC,'R','O','M','E',SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

CAIRO:
  DB SPC,SPC,'C','A','I','R','O',SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

MOSCOW:
  DB SPC,SPC,'M','O','S','C','O','W',SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

HONOLU:
  DB SPC,SPC,'H','O','N','O','L','U','L','U'
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

SYDNEY:
  DB SPC,SPC,'S','Y','D','N','E','Y',SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

PEKING:
  DB SPC,SPC,'P','E','K','I','N','G',SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
  DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC

```



```

HONG:
DB      SPC, SPC, 'H', 'O', 'N', 'G', SPC, 'K', 'O', 'N'
DB      'G', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

NEWDEL:
DB      SPC, SPC, 'N', 'E', 'W', SPC, 'D', 'E', 'L', 'H'
DB      'I', SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

KUWAIT:
DB      SPC, SPC, 'K', 'U', 'W', 'A', 'I', 'T', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

;*****
;      初期設定LCD表示データ
;*****
SET_D:
DB      'S', 'E', 'T', SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DATE_D:
DB      SPC, SPC, 'D', 'A', 'T', 'E', SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

TIME_D:
DB      SPC, SPC, 'T', 'I', 'M', 'E', SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETDATE:
DB      'S', 'E', 'T', SPC, 'D', 'A', 'T', 'E', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

YEAR:
DB      SPC, SPC, 'Y', 'E', 'A', 'R', QUEST, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MONTH:
DB      SPC, SPC, 'M', 'O', 'N', 'T', 'H', QUEST, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

DAY:
DB      SPC, SPC, 'D', 'A', 'Y', QUEST, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SETTIME:
DB      'S', 'E', 'T', SPC, 'T', 'I', 'M', 'E', SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

HOUR:
DB      SPC, SPC, 'H', 'O', 'U', 'R', QUEST, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

MINUTE:
DB      SPC, SPC, 'M', 'I', 'N', 'U', 'T', 'E', QUEST, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC

SECOND:
DB      SPC, SPC, 'S', 'E', 'C', 'O', 'N', 'D', QUEST, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
DB      SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC, SPC
    
```

```

;*****
;      通信処理表示データ
    
```

:*****

COMMUN:

```
DB 'C','O','M','M','U','N','I','C','A','T'
DB 'I','O','N',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

TRANS:

```
DB SPC,SPC,'T','R','A','N','S','M','I','T'
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

RECEIVE:

```
DB SPC,SPC,'R','E','C','E','I','V','E',SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

PC9801:

```
DB SPC,SPC,'P','C',SUB_SIG,'9','8','0','1',PAR_LE
DB 'R','S',SUB_SIG,'2','3','2','C',PAR_RI,SPC,SPC
DB SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
```

ELE_POC:

```
DB SPC,SPC,'E','L','E','C','T','R','O','N'
DB 'I','C',SPC,'P','O','C','K','E','T','B'
DB 'O','O','K',SPC,SPC,SPC,SPC,SPC,SPC,SPC,SPC
END
```

〔メモ〕

付 録 SPDチャートの説明

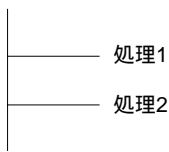

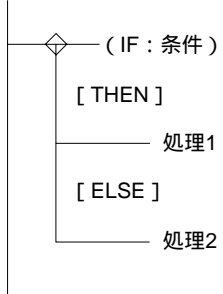
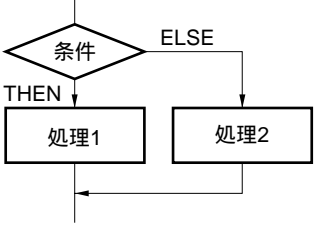

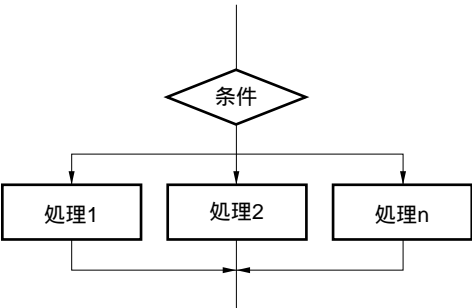
SPDとは、Structured Programming Diagramsの頭文字をとったもので、文字どおり訳せば「構造化プログラム図」と言えます。

構造化というのは、プログラムの論理処理の構造化のことで、論理の基本構造を用いて論理の設計、組み立てを行うことです。

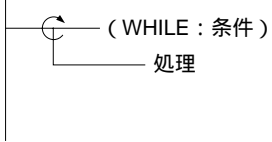
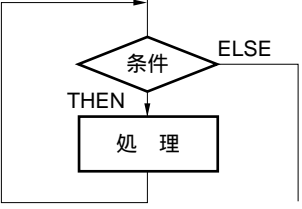
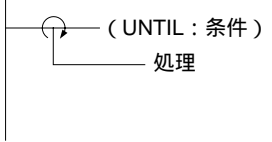
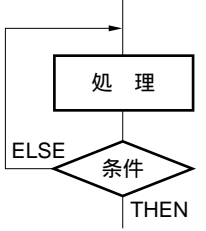
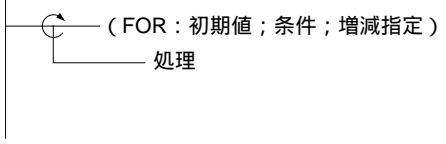
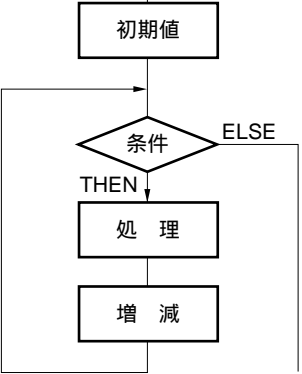
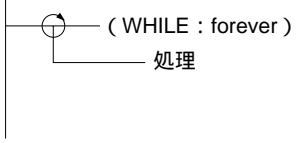
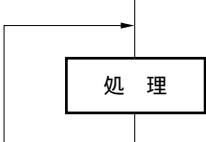
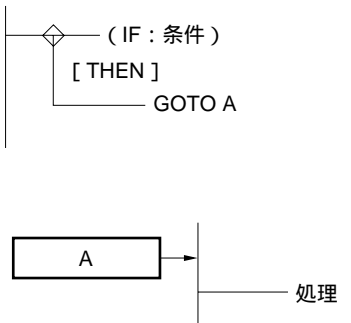
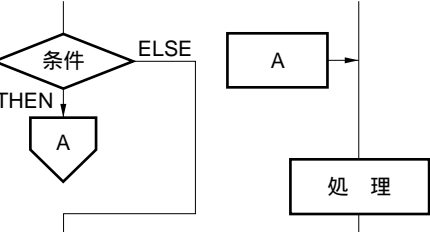
すべてのプログラムは、論理の基本構造（順次、選択、繰り返し）の組み合わせのみで作成することができ（これを構造化定理といいます）、構造化することでプログラムの流れが明確になり、信頼性が向上します、プログラムの構造化を表現するには、いろいろな方法がありますが、当社では、SPDという図式化技法を用いています。

以下に、SPD技法で用いるSPD記号の説明、およびフロー・チャート記号との対比を示します。

表付 - 1 SPD記号とフロー・チャートの対比 (1/2)

処理名称	SPD記号	フロー・チャート記号
順次処理		
条件分岐 (IF)		
条件分岐 (SWITCH)		

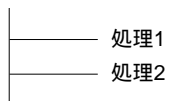
表付 - 1 SPD記号とフロー・チャートの対比 (2/2)

処理名称	SPD記号	フロー・チャート記号
条件ループ (WHILE)	 <p>(WHILE : 条件) 処理</p>	
条件ループ (UNTIL)	 <p>(UNTIL : 条件) 処理</p>	
条件ループ (FOR)	 <p>(FOR : 初期値 ; 条件 ; 増減指定) 処理</p>	
無限ループ	 <p>(WHILE : forever) 処理</p>	
結合子	 <p>(IF : 条件) [THEN] GOTO A</p> <p>A</p> <p>処理</p>	

1 . 順次処理

順次処理は、処理を上から下へ出現順に実行します。

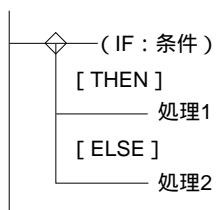
• SPDチャート



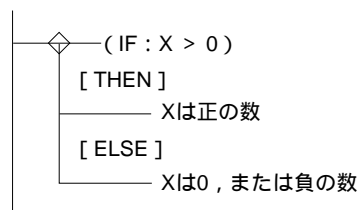
2 . 条件分岐：2分岐（IF）

IFに示した条件の真偽（THEN/ELSE）により処理内容を選択します。

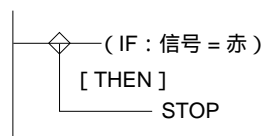
• SPDチャート



例 1 . Xの正負判別



2 . 信号が赤ならSTOPする

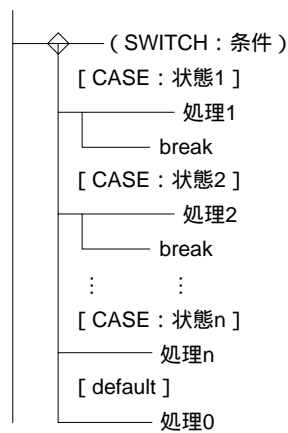


3 . 条件分岐：多分岐（SWITCH）

SWITCHに示した条件を，CASEで示された状態と比較し，処理を選択します。SWITCH文の処理は，一致した状態のみの処理を実行する場合と，一致した状態から下へ処理を続ける場合の二通りがあります（処理が下へ続かない場合は，'break' を記述）。また，一致した状態がない場合は，'default' の処理を実行します（'default' の記述は任意）。

（1）一致した状態のみの場合

・SPDチャート



（実行内容）

状態1のとき：処理1

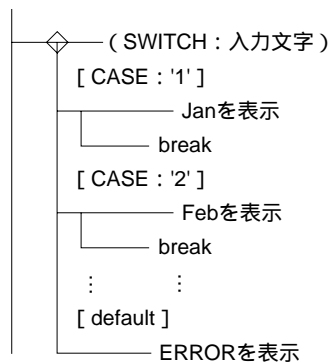
状態2のとき：処理2

⋮
⋮

状態nのとき：処理n

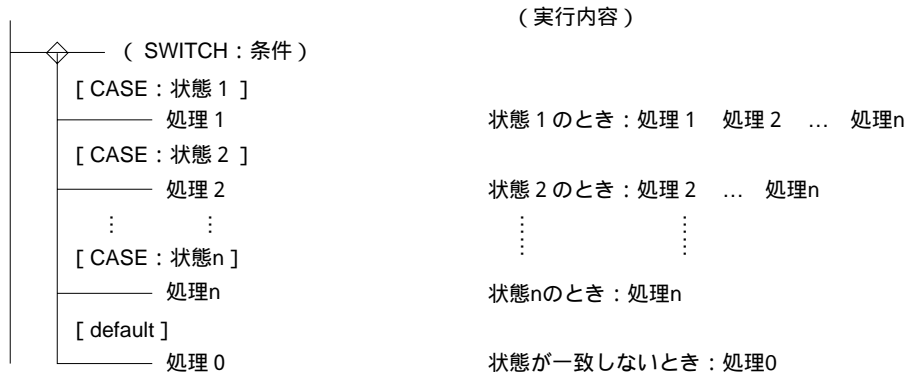
状態が一致しないとき：処理0

例 入力文字により，月名を表示する

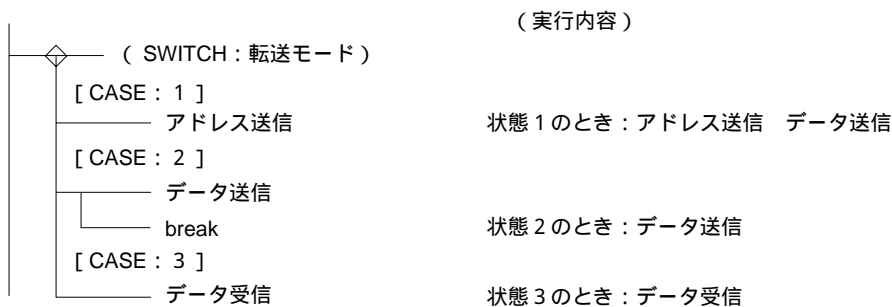


(2) 一致した状態から処理が続く場合

・SPDチャート



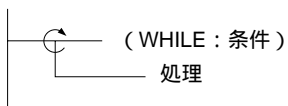
例 シリアル・インタフェースの送受信



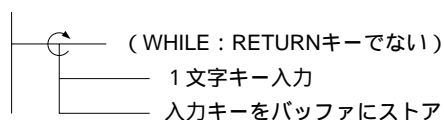
4 . 条件ループ (WHILE)

WHILEに示した条件を判定し、条件が成立している間、処理を繰り返し実行します（初めから条件が不成立の場合は、処理を実行しません）。

・SPDチャート



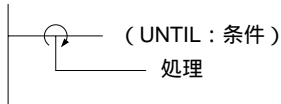
例 RETURNキー入力があるまで、キーをバッファリングする



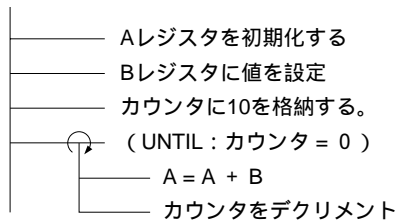
5 . 条件ループ (UNTIL)

処理を行ったあとにUNTILに示した条件を判定し、条件が成立するまで処理を繰り返し実行します（初めから条件が不成立の場合でも、処理を一度実行します）。

・SPDチャート



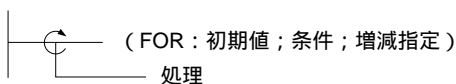
例 Bレジスタの値を10倍してAレジスタに格納する



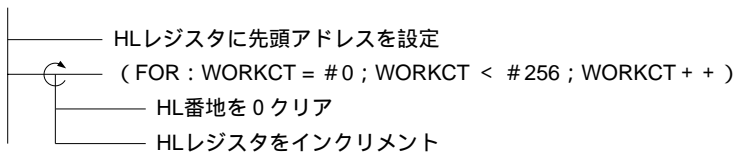
6 . 条件ループ (FOR)

FORに示されたパラメータの条件が成立している間、処理を繰り返し実行します。

・SPDチャート



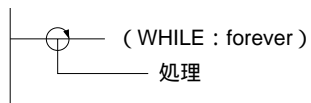
例 HL番地から256バイトを0クリアする



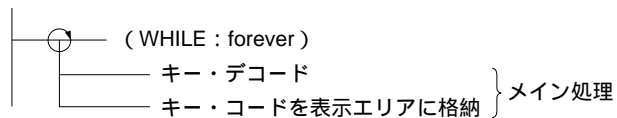
7 . 無限ループ

WHILEの条件として'forever'を設定すると、処理を無限に繰り返し実行します。

・SPDチャート



例 メイン処理を繰り返し実行する

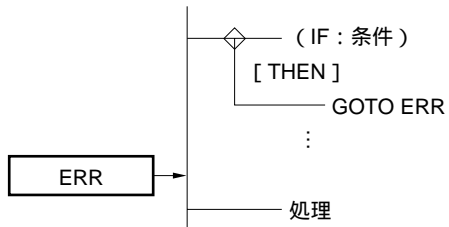


8 . 結合子 (GOTO)

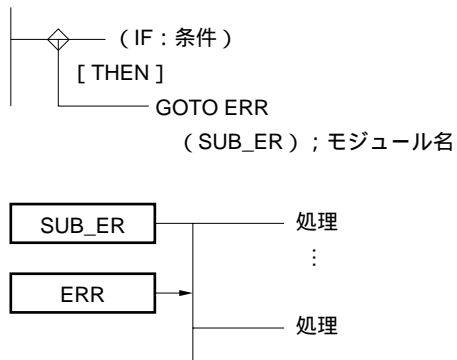
無条件に指定されたアドレスに分岐します。

・SPDチャート

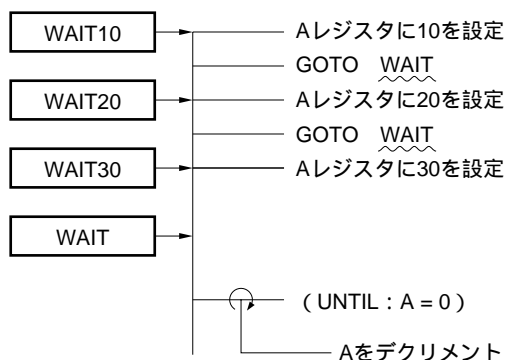
(1) 同じモジュールに分岐



(2) 異なるモジュールに分岐



例 サブルーチンの開始アドレスで、パラメータを選択しウエイトを設定する



9 . 結合子 (継続)

1 モジュールのSPDが、複数のページにおよぶ場合に使用し、処理の流れを示します。

・SPDチャート



〔メ モ〕

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494
FAX : 044-435-9608
E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東京 (03)3798-6106, 6107,
6108

名古屋 (052)222-2375

大阪 (06)6945-3178, 3200,
3208, 3212

仙台 (022)267-8740

郡山 (024)923-5591

千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,
6112

立川 (042)526-5981, 6167

松本 (0263)35-1662

静岡 (054)254-4794

金沢 (076)232-7303

松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156

水戸 (029)226-1702

広島 (082)242-5504

高崎 (027)326-1303

鳥取 (0857)27-5313

太田 (0276)46-4014

名古屋 (052)222-2170, 2190

福岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【インターネット電子デバイス・ニュース】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

アンケート記入のお願い

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

[ドキュメント名] μPD78014サブシリーズ アプリケーション・ノート 電子手帳編
(U12843JJ1V2AN00 (第1版))

[お名前など] (さしつかえのない範囲で)
御社名(学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価 (各欄に をご記入ください)

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

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは
NEC販売員, 特約店販売員, その他 ()

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

キ
コ
ン
テ
ン
ト