

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

μ PD6133シリーズ

リモコン送信プログラム例

	概 説	第1編
	赤外線リモコンの基礎	1
	リモコン送信プログラムの概要	2
	48キー対応のプログラム	第2編
	ハードウェア構成	1
	送信波形	2
	タイミング・チャート	3
	出力コード	4
	ソフトウェア構成	5
	プログラム説明	6
	プログラム変更時の注意点	7
	プログラム・リスト	8
	80キー対応のプログラム	第3編
	ハードウェア構成	1
	送信波形	2
	タイミング・チャート	3
	出力コード	4
	ソフトウェア構成	5
	プログラム説明	6
	プログラム変更時の注意点	7
	本プログラム使用上の注意点	8
	プログラム・リスト	9

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

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

本資料の内容は、後日変更する場合があります。

文書による当社の承諾なしに本資料の転載複製を禁じます。

本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的所有権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。

当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。

当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。

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

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

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

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

この製品は耐放射線設計をしておりません。

巻末にアンケート・コーナ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。

(メ モ)

はじめに

- 対象者** このアプリケーション・ノートは、赤外線リモート・コントロール送信機用のマイクロコントローラ、 μ PD6133シリーズの機能を理解し、それを用いたアプリケーション・プログラムを設計するエンジニアを対象とします。
- 目的** このアプリケーション・ノートは、 μ PD6133シリーズの持つ機能を、応用プログラム例を用いてユーザに理解していただくことを目的としています。
- 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。

概説

- ・赤外線リモコンの基礎
- ・リモコン送信プログラムの概要

48キー対応のプログラム

- ・ハードウェア構成
- ・送信波形
- ・タイミング・チャート
- ・出力コード
- ・ソフトウェア構成
- ・プログラム説明
- ・プログラム変更時の注意点
- ・プログラム・リスト

80キー対応のプログラム

- ・ハードウェア構成
- ・送信波形
- ・タイミング・チャート
- ・出力コード
- ・ソフトウェア構成
- ・プログラム説明
- ・プログラム変更時の注意点
- ・本プログラム使用上の注意点
- ・プログラム・リスト

- 凡 例
- データ表記の重み : 左が上位桁, 右が下位桁
 - 注 : 本文中につけた注の説明
 - 注意 : 気をつ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... x x x x B
10進数... x x x x
16進数... x x x x H

関連資料 次の資料をあわせてご利用ください。
 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示はしていません。あらかじめご了承ください。

資 料	資 料 番 号	
	和 文	英 文
μPD6133, 6134 データ・シート	U10454J	U10454E
μPD6604 データ・シート	U11281J	U11281E
μPD63, 64 データ・シート	U11371J	U11371E
SM6133 バージョン1.0 ユーザーズ・マニュアル	U11128J	U11128E
AS6133 アセンブラ ユーザーズ・マニュアル	U10115J	U10115E

ご 注 意 記載のプログラム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。

目 次

第1編 概 説	...	1
第1章 赤外線リモコンの基礎	...	3
第2章 リモコン送信プログラムの概要	...	5
2.1 ジェネラル・フロー・チャート	...	5
2.2 処理概要	...	5
2.2.1 初期設定処理	...	5
2.2.2 STOPモード	...	5
2.2.3 キー入力処理	...	6
2.2.4 送信処理	...	6
第2編 48キー対応のプログラム	...	7
第1章 ハードウェア構成	...	9
1.1 応用回路例	...	9
1.2 キー・マトリクス	...	10
第2章 送信波形	...	11
2.1 NEC-Rフォーマット	...	11
第3章 タイミング・チャート	...	13
3.1 キー入力-REM出力タイミング・チャート	...	13
3.1.1 キーの押し続け時のタイミング	...	13
3.1.2 キー・オフ時のタイミング	...	14
3.2 キー動作のタイミング・チャート	...	17
3.2.1 キー確定までの出力パターン	...	17
3.2.2 二重押しキーの動作	...	23
3.2.3 キーの渡り操作	...	25
第4章 出力コード	...	27
第5章 ソフトウェア構成	...	29
5.1 ジェネラル・フロー・チャート	...	29
5.2 プログラム・メモリ(ROM)構成	...	30
5.3 データ・メモリ(RAM)構成	...	32
5.4 フラグ・マップ	...	38

第6章	プログラム説明	...	39
6.1	初期値設定	...	39
6.1.1	処理説明	...	39
6.1.2	ディテール・フロー・チャート	...	43
6.2	キー入力処理	...	45
6.2.1	処理説明	...	45
6.2.2	ディテール・フロー・チャート	...	52
6.3	送信処理	...	65
6.3.1	処理説明	...	66
6.3.2	ディテール・フロー・チャート	...	71
第7章	プログラム変更時の注意点	...	83
第8章	プログラム・リスト	...	85
第3編	80キー対応のプログラム	...	101
第1章	ハードウェア構成	...	103
1.1	応用回路例	...	103
1.2	キー・マトリクス	...	104
第2章	送信波形	...	105
2.1	NEC-Rフォーマット	...	105
第3章	タイミング・チャート	...	107
3.1	キー入力-REM出力タイミング・チャート	...	107
3.1.1	キーの押し続け時のタイミング	...	107
3.1.2	キー・オフ時のタイミング	...	108
3.2	キー動作のタイミング・チャート	...	111
3.2.1	キー確定までの出力パターン	...	111
3.2.2	二重押しキーの動作	...	118
3.2.3	キーの渡り操作	...	120
第4章	出力コード	...	121
第5章	ソフトウェア構成	...	123
5.1	ジェネラル・フロー・チャート	...	123
5.2	プログラム・メモリ(ROM)構成	...	124
5.3	データ・メモリ(RAM)構成	...	126
5.4	フラグ・マップ	...	132

第 6 章	プログラム説明	...	133
6.1	初期値設定	...	133
6.1.1	処理説明	...	133
6.1.2	ディテール・フロー・チャート	...	137
6.2	キー入力処理	...	139
6.2.1	処理説明	...	139
6.2.2	ディテール・フロー・チャート	...	146
6.3	送信処理	...	158
6.3.1	処理説明	...	159
6.3.2	ディテール・フロー・チャート	...	164
第 7 章	プログラム変更時の注意点	...	175
第 8 章	本プログラム使用上の注意点	...	177
第 9 章	プログラム・リスト	...	179

図の目次 (1/2)

図番号

タイトル, ページ

第1編 概 説

- 1 - 1 赤外線リモコンの構成例 ... 3
- 1 - 2 NECフォーマットの送信波形例 ... 4

第2編 48キー対応のプログラム

- 1 - 1 応用回路例 ... 9
- 1 - 2 キー・マトリクス ... 10

- 2 - 1 NEC-Rフォーマット ... 11

- 3 - 1 キーを押し続けたときのタイミング・チャート ... 14
- 3 - 2 キー・オフ時のタイミング・チャート ... 15
- 3 - 3 キー確定までの出力パターン ... 18
- 3 - 4 二重押しキー動作のタイミング・チャート ... 23
- 3 - 5 キーの渡り操作時の動作タイミング例 ... 25

- 5 - 1 ジェネラル・フロー・チャート ... 29
- 5 - 2 ROMマップ ... 30
- 5 - 3 データ・テーブル・マップ ... 31

- 6 - 1 キー入力信号のチャタリング ... 45
- 6 - 2 チャタリング発生時のキー・オン判定 ... 46
- 6 - 3 48キー・マトリクス ... 47
- 6 - 4 キー・マトリクス ... 48
- 6 - 5 キー・データ(キー位置) ... 48
- 6 - 6 K₁₀-K₁₃に入力があったときのキー・データのビット構成 ... 49
- 6 - 7 S₀, S₁に入力があったときのキー・データのビット構成 ... 50
- 6 - 8 データ・ポインタの構成 ... 50
- 6 - 9 タイマ・カウンタの構成 ... 65
- 6 - 10 タイマ・カウンタの時間管理 ... 65
- 6 - 11 フレーム・スペース送信時間の分割方法 ... 69
- 6 - 12 ビット・データ送信時のキー・オフ・チェック ... 70

図の目次 (2/2)

図番号

タイトル, ページ

第3編 80キー対応のプログラム

1 - 1	応用回路例 ...	103
1 - 2	キー・マトリクス ...	104
2 - 1	NEC-Rフォーマット ...	105
3 - 1	キーを押し続けたときのタイミング・チャート ...	108
3 - 2	キー・オフ時のタイミング・チャート ...	109
3 - 3	キー確定までの出力パターン ...	112
3 - 4	二重押しキー動作のタイミング・チャート ...	118
3 - 5	キーの渡り操作時の動作タイミング例 ...	120
5 - 1	ジェネラル・フロー・チャート ...	123
5 - 2	ROMマップ ...	124
5 - 3	データ・テーブル・マップ ...	125
6 - 1	キー入力信号のチャタリング ...	139
6 - 2	チャタリング発生時のキー・オン判定 ...	140
6 - 3	80キー・マトリクス ...	141
6 - 4	キー・マトリクス ...	142
6 - 5	キー・データ(キー位置) ...	142
6 - 6	K10-K13に入力があつたときのキー・データのビット構成 ...	143
6 - 7	S1に入力があつたときのキー・データのビット構成 ...	144
6 - 8	データ・ポインタの構成 ...	144
6 - 9	タイマ・カウンタの構成 ...	158
6 - 10	タイマ・カウンタの時間管理 ...	158
6 - 11	フレーム・スペース送信時間の分割方法 ...	162
6 - 12	ビット・データ送信時のキー・オフ・チェック ...	163

表の目次

表番号	タイトル, ページ
第2編 48キー対応のプログラム	
1 - 1	端子の機能 ... 10
4 - 1	出力コード ... 27
5 - 1	RAMマップ ... 32
5 - 2	キー入力処理時使用RAMマップ ... 33
5 - 3	送信処理時使用RAMマップ ... 34
5 - 4	RAM使用説明 ... 35
5 - 5	フラグ・マップ ... 38
6 - 1	コントロール・レジスタ0 (P3) ... 40
6 - 2	タイマ・クロックとキャリア周波数の設定 ... 40
6 - 3	コントロール・レジスタ1 (P4) ... 41
6 - 4	HALT命令の解除条件一覧表 ... 42
6 - 5	データ1送信カウンタ (R9) の内訳 ... 66
6 - 6	‘データ1’の個数に対するフレーム・スペースの時間 ... 69
第3編 80キー対応のプログラム	
1 - 1	端子の機能 ... 104
4 - 1	出力コード ... 121
5 - 1	RAMマップ ... 126
5 - 2	キー入力処理時使用RAMマップ ... 127
5 - 3	送信処理時使用RAMマップ ... 128
5 - 4	RAM使用説明 ... 129
5 - 5	フラグ・マップ ... 132
6 - 1	コントロール・レジスタ0 (P3) ... 134
6 - 2	タイマ・クロックとキャリア周波数の設定 ... 134
6 - 3	コントロール・レジスタ1 (P4) ... 135
6 - 4	HALT命令の解除条件一覧表 ... 136
6 - 5	データ1送信カウンタ (R9) の内訳 ... 159
6 - 6	‘データ1’の個数に対するフレーム・スペースの時間 ... 162

第1編 概 説

(メ モ)

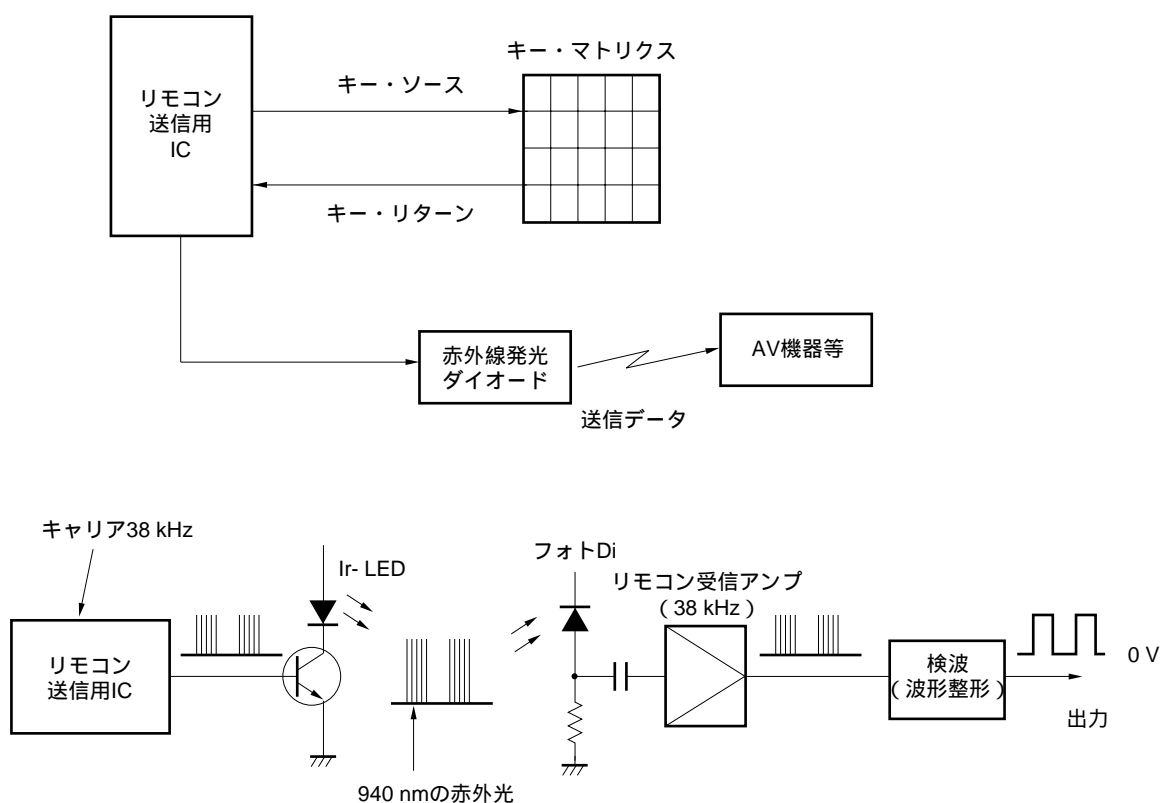
第1章 赤外線リモコンの基礎

赤外線リモートコントロール送信機（以下，“赤外線リモコン”または単に“リモコン”と略す）は赤外線の間断を行うことによって，‘0’，‘1’のデータを送信し，各種の電子制御機器の遠隔制御を行います。

赤外線リモコンは送信用素子として，赤外線発光ダイオード（Ir-LED）を使用することにより，リモコン送信機が小型化しやすく，安価に製作ができます。

次に一般的な赤外線リモコンの構成を示します。

図1-1 赤外線リモコンの構成例



赤外線リモコンの有効距離を7～10 m程度確保するためにIr-LEDに流す電流は，使用するLEDにもよりますが，だいたい300 mA～500 mAです。ただし，この電流では電池の寿命が短くなるため，変調をかけて断続を行い，平均電流をピーク電流の1/10以下に落としています。

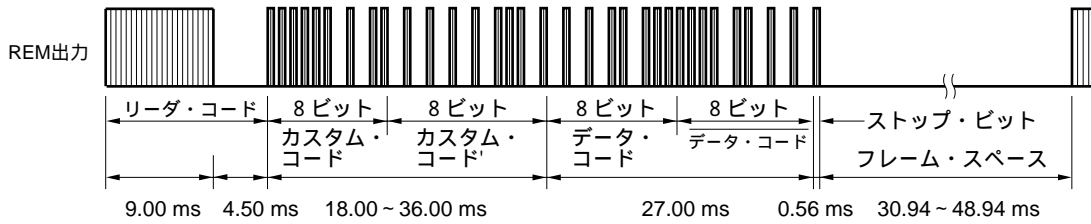
Ir-LEDより出力された赤外光は，受信機のフォト・ダイオードで再び電気信号に変換されます。

この受信信号は非常に微弱なため，受信アンプで増幅したのち，検波されて，‘0’，‘1’の信号に復調されます。

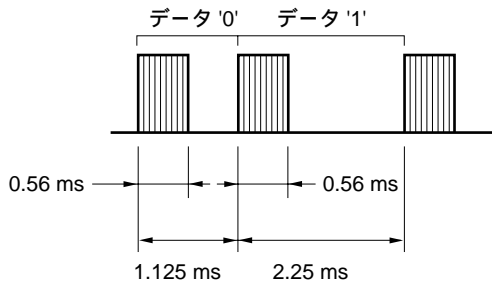
赤外線リモコンの送信データの例として、NECフォーマットの送信波形を次に示します。
Ir-LEDは、ハイ・レベル期間に発光、ロウ・レベル期間に消灯します。

図1-2 NECフォーマットの送信波形例

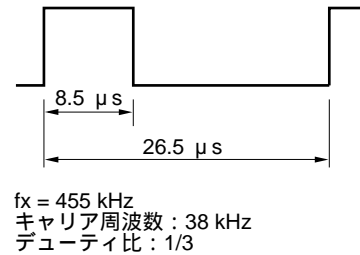
(1) REM出力波形



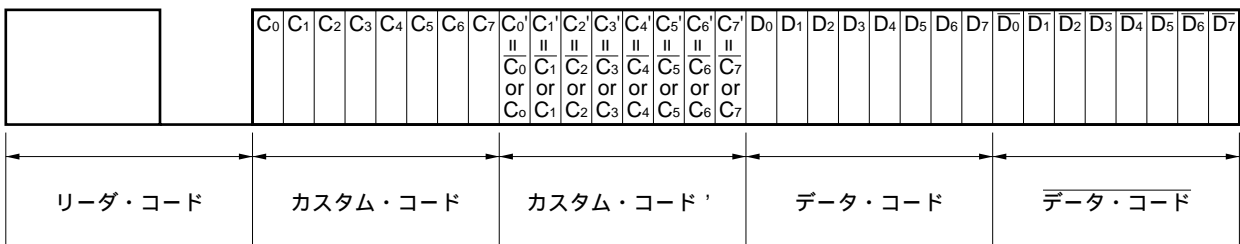
(2) ビット・データ形式



(3) キャリア波形

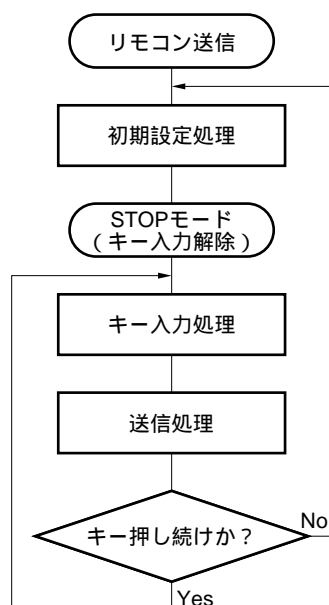


(4) 各コードのビット並び



第2章 リモコン送信プログラムの概要

2.1 ジェネラル・フロー・チャート



2.2 処理概要

2.2.1 初期設定処理

一般的にリモコンは、電源に電池を使用するため、Ir-LEDが発光したとき電源電圧の変動によって、RAMやポートに保持している内容が突然変化する場合を想定しておく必要があります。

そのため、リモコン信号を送信するごとに、RAMおよびポートの初期設定処理を実行するようにプログラムを組みます。

初期設定処理の設定内容は、次のとおりです。

- ・ポート設定、キャリアの周波数/デューティ比、キャリアの有無
- ・キャリア周波数とタイマ・クロックの分周比の設定
- ・各ポートの入出力モード設定および、プルダウン抵抗の有無

2.2.2 STOPモード

STOPモードは、プログラムの実行を停止するとともに、システム・クロックの発振を停止することによって低消費電流を実現するためのモードです。

リモコンのプログラムの場合では、キー入力を解除条件とするSTOPモードを設定することによって、キーが押されていないとき（スタンバイ状態）の消費電流を低減することができます。

2.2.3 キー入力処理

キー入力処理では、次の処理をします。

- ・キー・スキャンして、どのキーが押されたかを判別します。
- ・確実なキー操作を確認します（チャタリング除去）。
- ・押されたキーに対応するキー・データを作成し、RAMに格納します。

2.2.4 送信処理

送信処理では、次の処理をします。

- ・キー入力処理で作成したキー・データに対応する送信データを、各メーカーのリモコン送信フォーマットに従って送信します。
- ・1回目の送信終了後、キーがまだ押し続けられているかどうかを判断します。
- ・キーが押し続けられていると判断した場合は、キー入力処理に移り、押されているキーに変化がないかどうかをチェックします。
- ・キーが離されたと判断した場合は、初期設定処理をして、新たにキーが押されるまでSTOPモードに入り、消費電力を抑えます。

第2編 48キー対応のプログラム

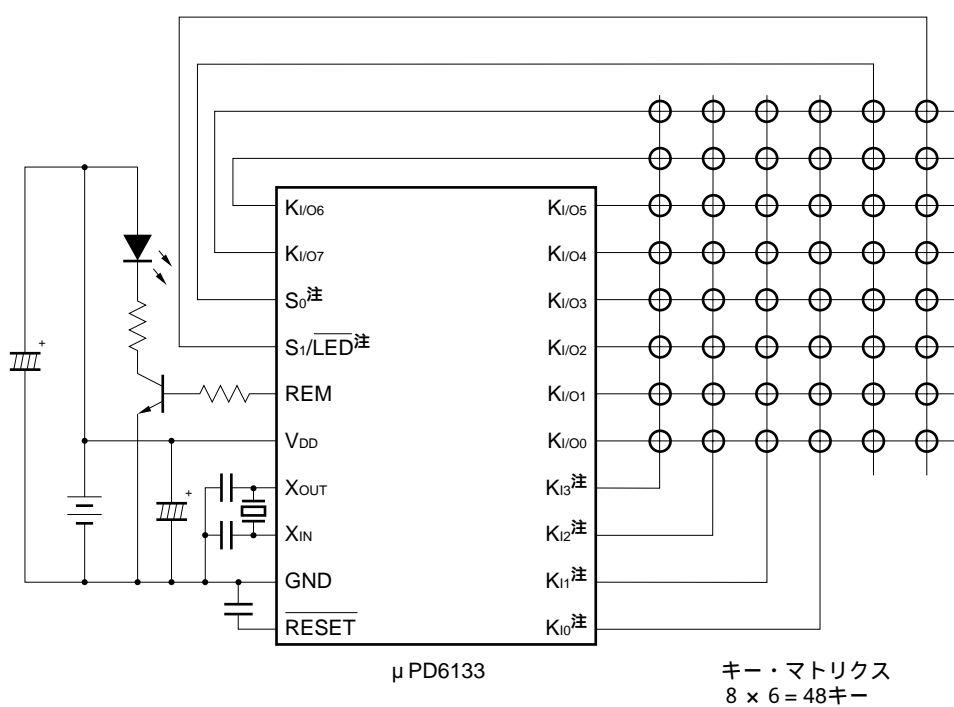
〔メ モ〕

第1章 ハードウェア構成

1.1 応用回路例

図1-1に応用回路例を示します。

図1-1 応用回路例



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

K1, K1/O, S0, S1/LED端子で48キーのキー・マトリクスを構成しています。送信コードはREM端子からキャリア付きで出力されます。

各端子の機能を表1-1に示します。

表1-1 端子の機能

端子名	入出力	機能
K _{1/00} -K _{1/07}	出力	キー・ソース (アクティブ・ハイ)
K ₁₀ -K ₁₃	入力	キー・リターン (アクティブ・ハイ)
S ₀	入力	
S ₁ /LED	入力	
REM	出力	赤外線リモコン信号 (キャリア付き)

1.2 キー・マトリクス

図1-2にキー・マトリクスを示します。

各キーのK_n (n=1~48) という記号は、キーの位置を示しています。

図1-2 キー・マトリクス

K _{1/07}	K29	K30	K31	K32	K47	K48
K _{1/06}	K25	K26	K27	K28	K45	K46
K _{1/05}	K21	K22	K23	K24	K43	K44
K _{1/04}	K17	K18	K19	K20	K41	K42
K _{1/03}	K13	K14	K15	K16	K39	K40
K _{1/02}	K9	K10	K11	K12	K37	K38
K _{1/01}	K5	K6	K7	K8	K35	K36
K _{1/00}	K1	K2	K3	K4	K33	K34
	K ₁₃	K ₁₂	K ₁₁	K ₁₀	S ₁	S ₀

第2章 送信波形

REM端子から出力される送信波形は、NEC連続送信（以下、NEC-Rと記す）フォーマットです。
出力データ・コードの内容については、第4章 出力コードを参照してください。

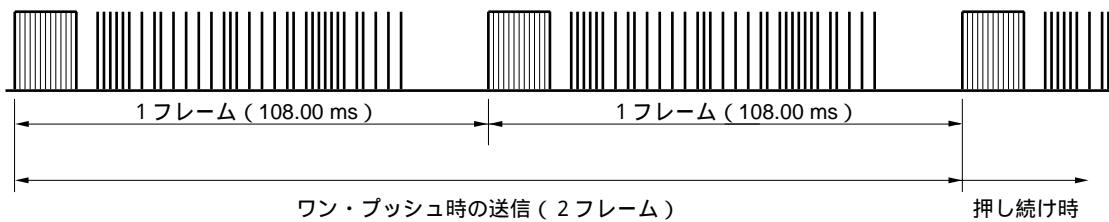
2.1 NEC-Rフォーマット

NEC-Rフォーマットを図2 - 1に示します。

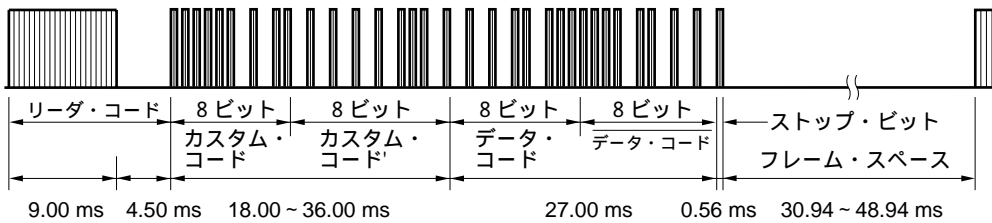
2

図2 - 1 NEC-Rフォーマット

・送信波形

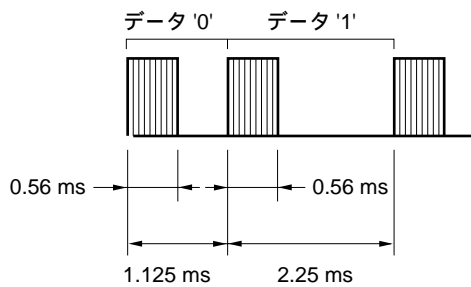


・1フレームの送信波形

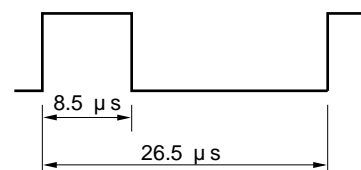


・2フレーム目以降の送信波形... 1フレーム目の送信波形と同様

・ビット・データ形式



・キャリアの波形



$f_x = 455 \text{ kHz}$
キャリア周波数: 38 kHz
デューティ比: 1/3

(メ モ)

第3章 タイミング・チャート

3.1 キー入力-REM出力タイミング・チャート

オン・チャタリング除去処理は、9.00 msごとにキー入力をチェックし、3回続けてオンと判定されたとき、キーがオンになったと判断します。また、オン・チャタリング除去処理ではキーの状態変化（キーのオン/オフ、キーの押し続け）をチェックしています。

次にオフ・チャタリング除去処理の方法を示します。

ビット・データのロウ・レベル出力中にキー・オフのチェックを行います。

1フレーム（108 ms）の送信中、REM端子からロウ・レベルを出力するタイミング（34回）で、10回キー入力をチェックして、

（1）10回連続でキー・オフと判定された場合

...キーがオフになったと判断します。

（2）10回のチェック中に1回でもキー・オンと判定された場合

...キー・オンと判定された時点で、チェック回数のカウンタをクリアし、さらに10回キー入力をチェックします。

（3）（1）と（2）のチェックでキーがオフになったと判断されなかった場合

...キーが押し続けられている（キーがオンである）と判断します。

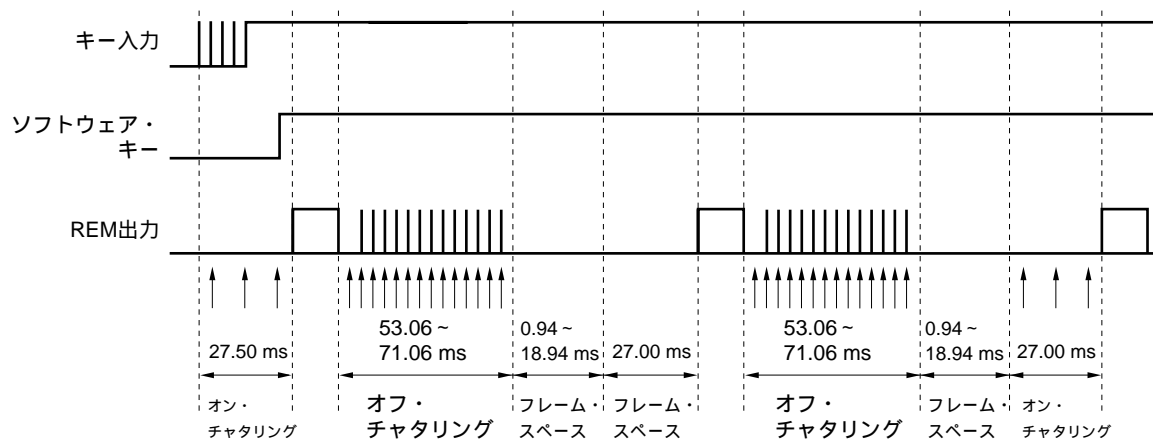
1フレーム目送信中にキーがオフになったと判断した場合でも、最低2フレームの送信を行ってから初期設定処理を行います。

3.1.1 キーの押し続け時のタイミング

図3 - 1にキーを押し続けた場合のタイミング・チャートを示します。

オン・チャタリング除去処理の詳細は、6.2.1（1）**チャタリング除去処理**を参考にしてください。

図3 - 1 キーを押し続けたときのタイミング・チャート



- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード, ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため, フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか, またはキー・オフが確定した場合, オン・チャタリング除去処理は行われずに, フレーム・スペース送信が行われます (= 27.00 ms)。

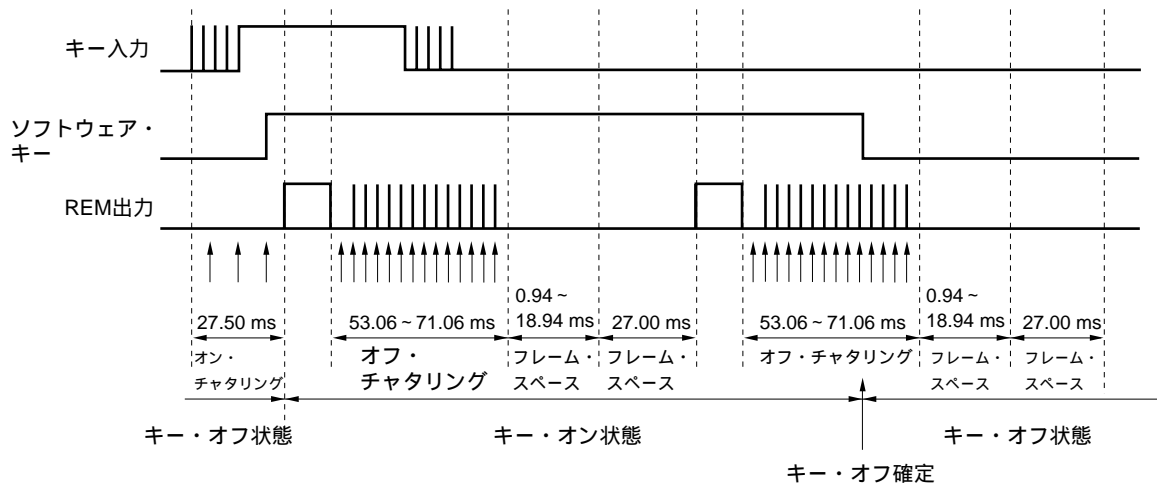
3.1.2 キー・オフ時のタイミング

図3 - 2にキーが送信途中で離された場合のタイミング・チャートを示します。

図3-2 キー・オフ時のタイミング・チャート (1/3)

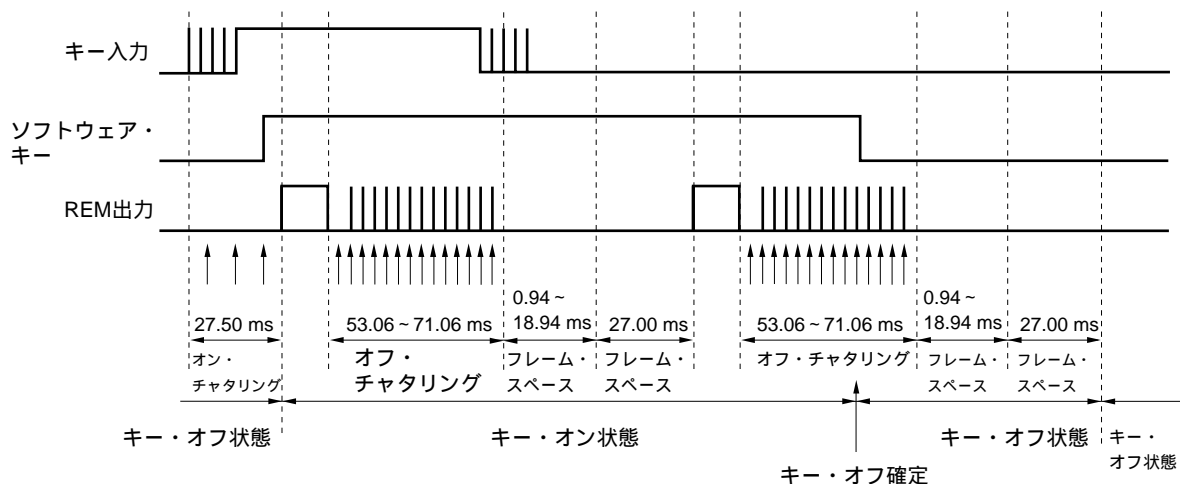
(1) 1フレーム目の送信中にキーが離された場合 (1フレーム目の送信中に、10回連続でキー・オフと判定された場合)

...キー・オフは、2フレーム目の送信中に確定します。



(2) 1フレーム目の送信中にキーが離された場合 (1フレーム目の送信中に、10回連続でキー・オフと判断されなかった場合)

...キー・オフは、2フレーム目の送信中に確定します。

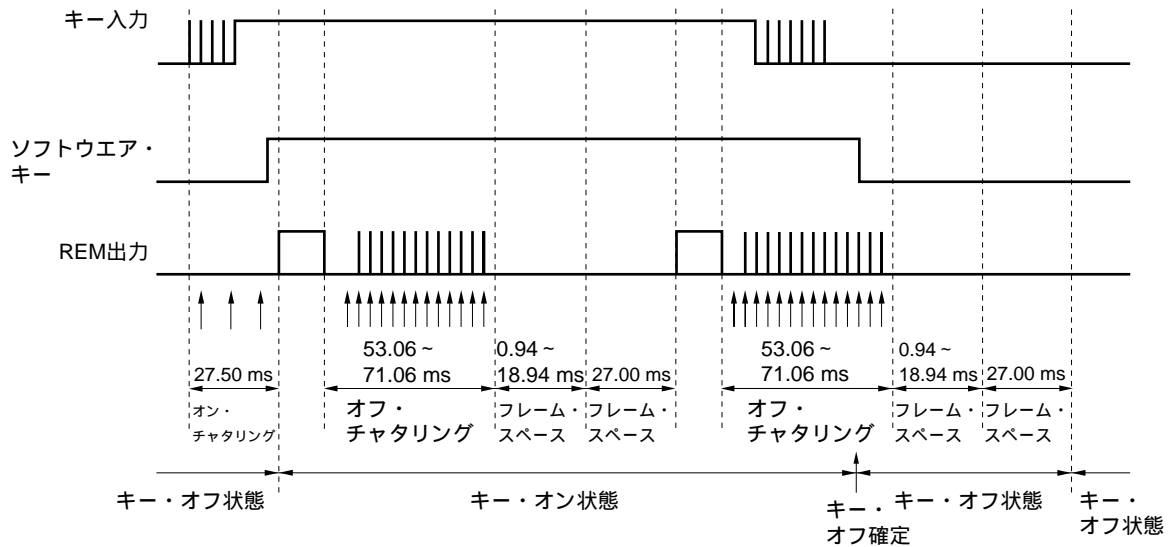


- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード、ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われず、フレーム・スペース送信が行われます (= 27.00 ms)。

図3 - 2 キー・オフ時のタイミング・チャート (2/3)

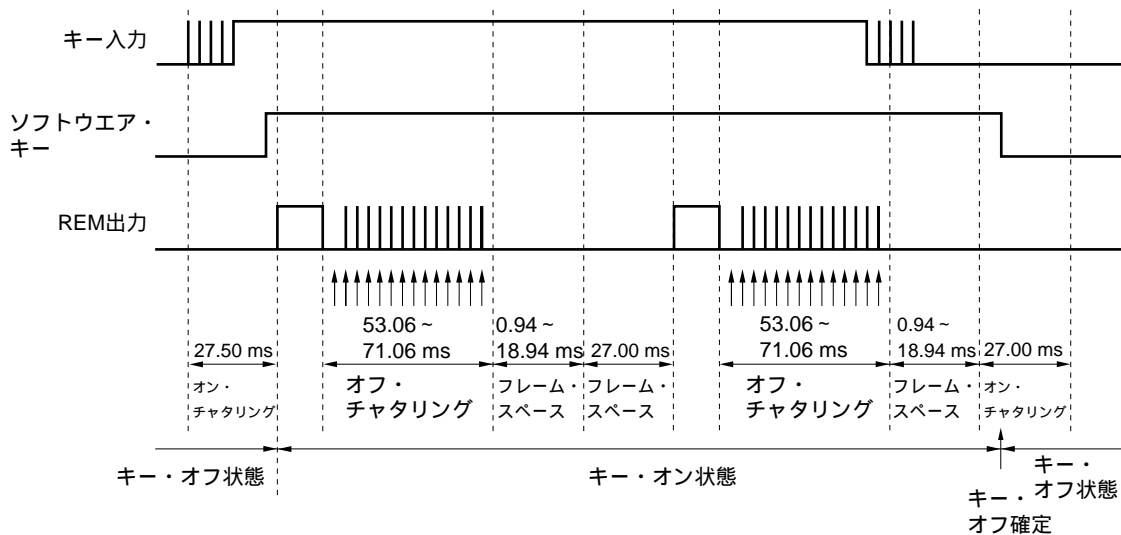
(3) 2フレーム目の送信中にキーが離された場合 (2フレーム目の送信中に、10回連続でキー・オフと判定された場合)

...キー・オフは、2フレーム目の送信中に確定します。



(4) 2フレーム目の送信中にキーが離された場合 (2フレーム目の送信中に、10回連続でキー・オフと判断されなかった場合)

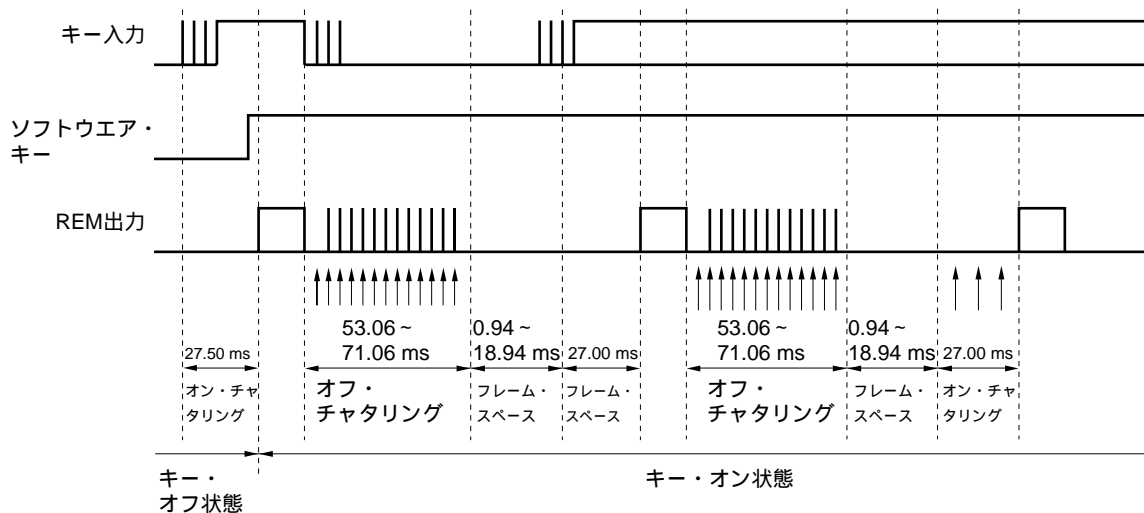
...キー・オフは、キーのオン・チャタリングで確定します。



- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード、ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われずに、フレーム・スペース送信が行われます (= 27.00 ms)。

図3 - 2 キー・オフ時のタイミング・チャート (3/3)

(5) 1フレーム目の送信途中でキーが離され、その後キーが押された場合
 ... 2フレーム目でキー・オンのため、キー・オフは確定しません。



- キー入力 : 実際のキー動作（手操作でキーを押す／離す）です。
- ソフトウェア・キー : ソフトウェア上でのキー動作（プログラム上でのキー入力判断）です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダ・コード、ストップ・ビットを含めたビット間（34ビット分）のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われずに、フレーム・スペース送信が行われます（= 27.00 ms）。

3.2 キー動作のタイミング・チャート

3.2.1 キー確定までの出力パターン

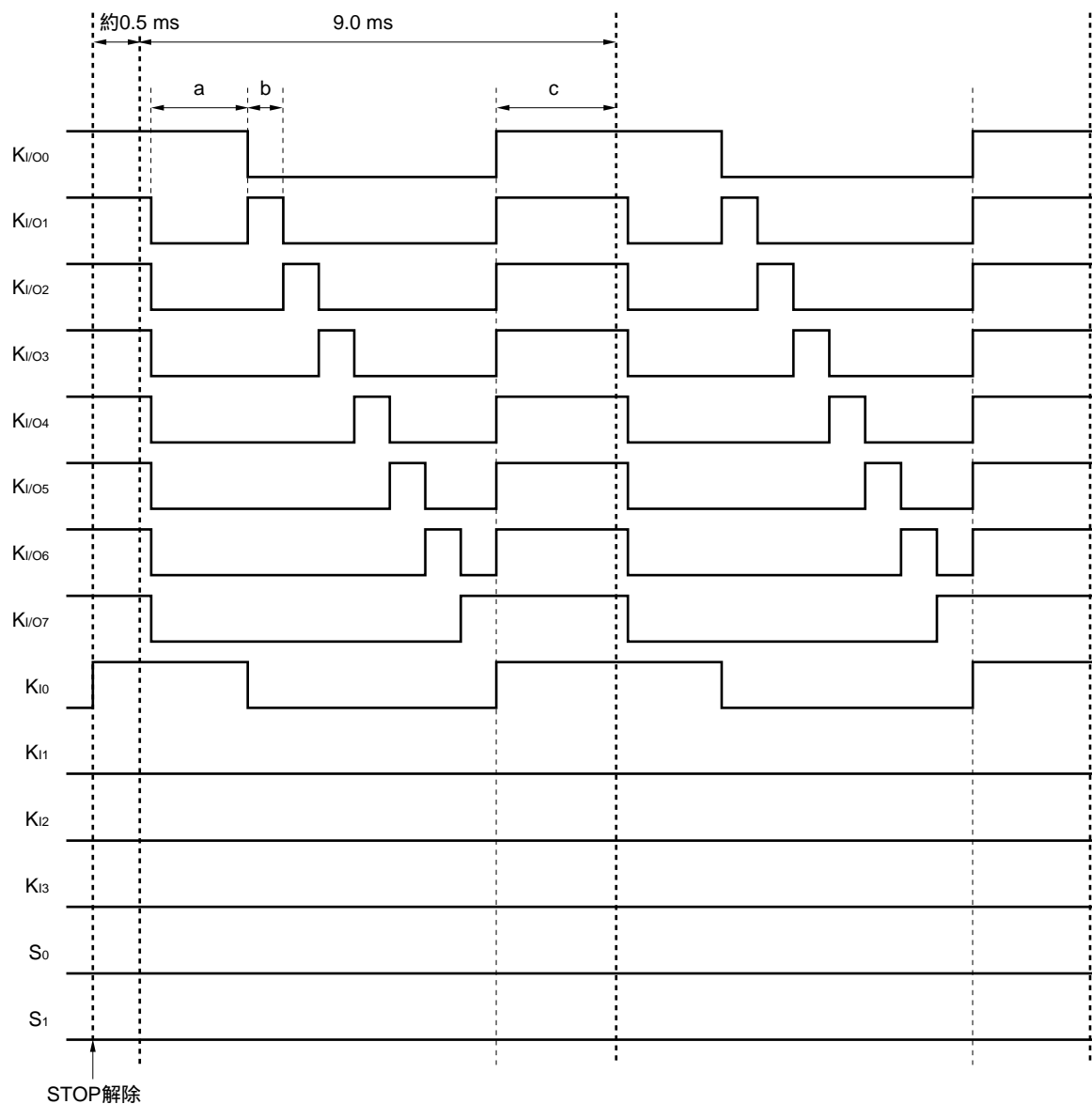
キー確定までには、オン・チャタリング除去処理（9.00 ms x 3回 = 27.00 ms）を行い、キーが3回同じ状態であればキーを確定します。

キー入力の確認は、キー・ソース出力端子をハイ・レベルに設定してから100 μs以上（fx = 455 kHz時、6命令以上）経過後に判断します。

キー確定までの詳細を、図3 - 3に示します。

図3 - 3 キー確定までの出力パターン (1/5)

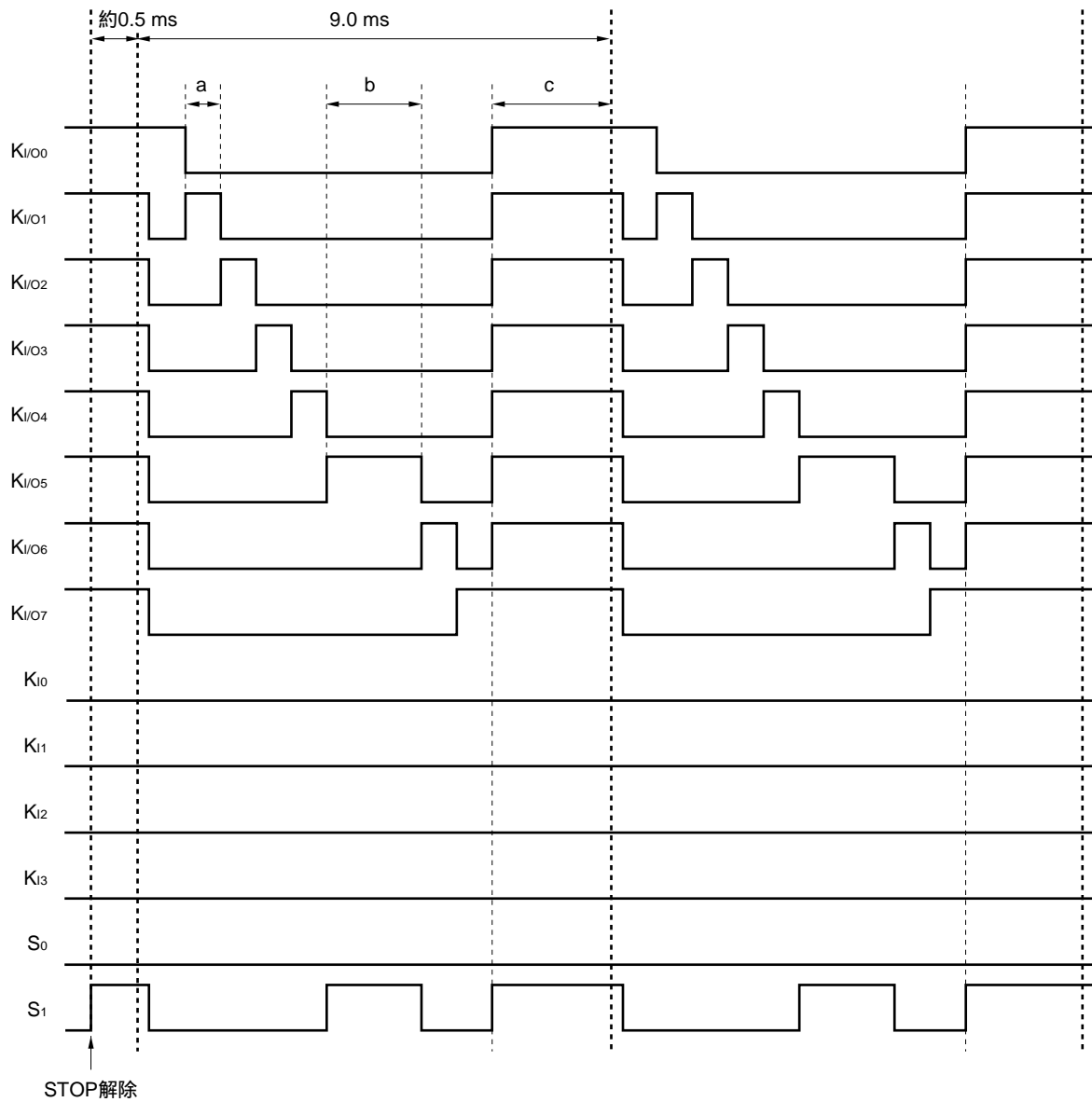
(1) K4が押された場合



- a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
実行時間は約1.34 ms ~ 1.95 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。
- c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。
実行時間は約4.23 ms ~ 4.84 msかかります。

図3 - 3 キー確定までの出力パターン (2/5)

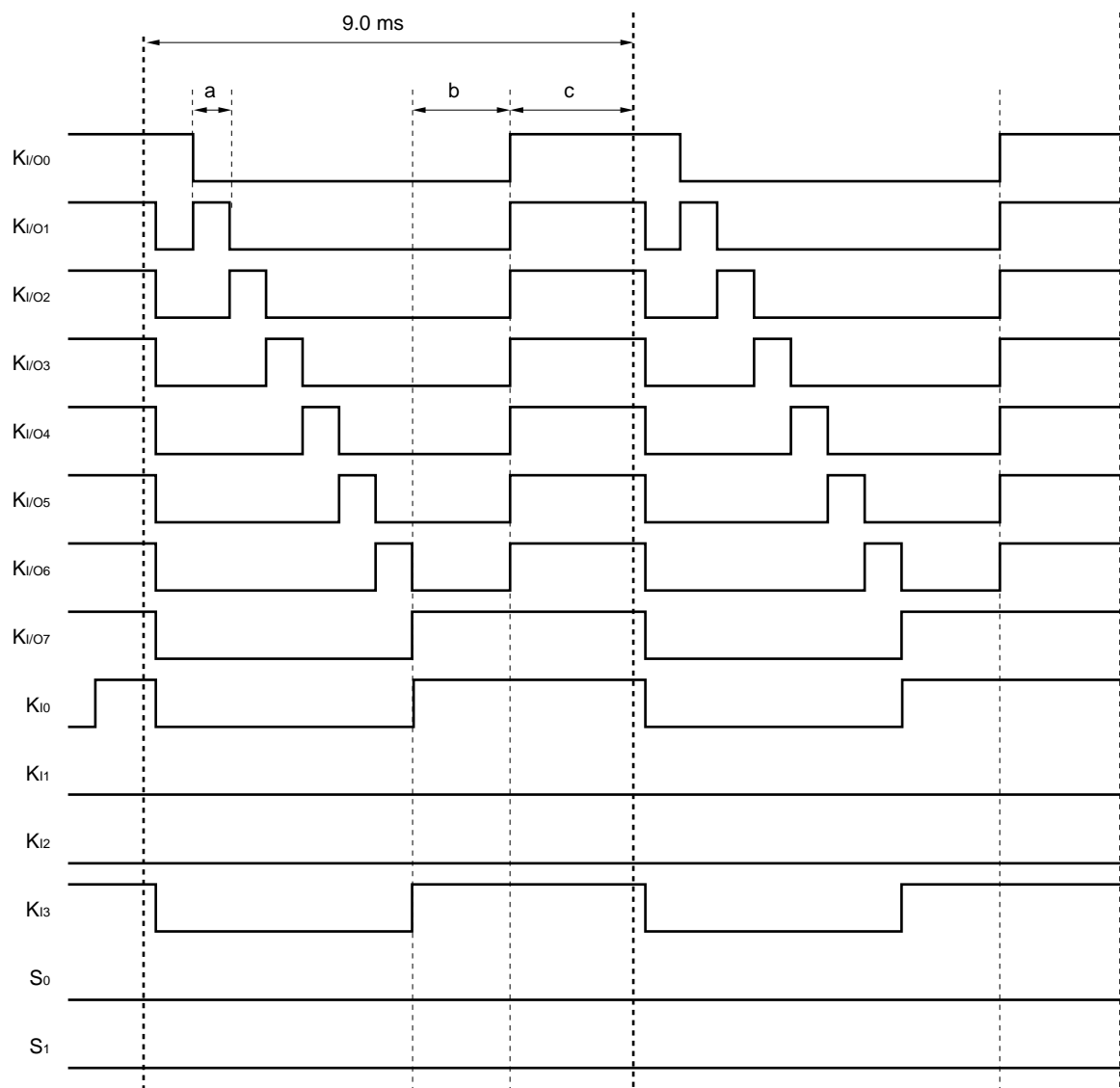
(2) K43が押された場合



- a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
実行時間は約1.34 ms ~ 1.95 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。
- c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。
実行時間は約4.23 ms ~ 4.84 msかかります。

図3 - 3 キー確定までの出力パターン (3/5)

(3) K29との有効な二重押しキー (K32) が押された場合 (K29 + K32)



a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。

実行時間は約1.34 ms ~ 1.95 msです。

b : キー入力のチェックを行います。

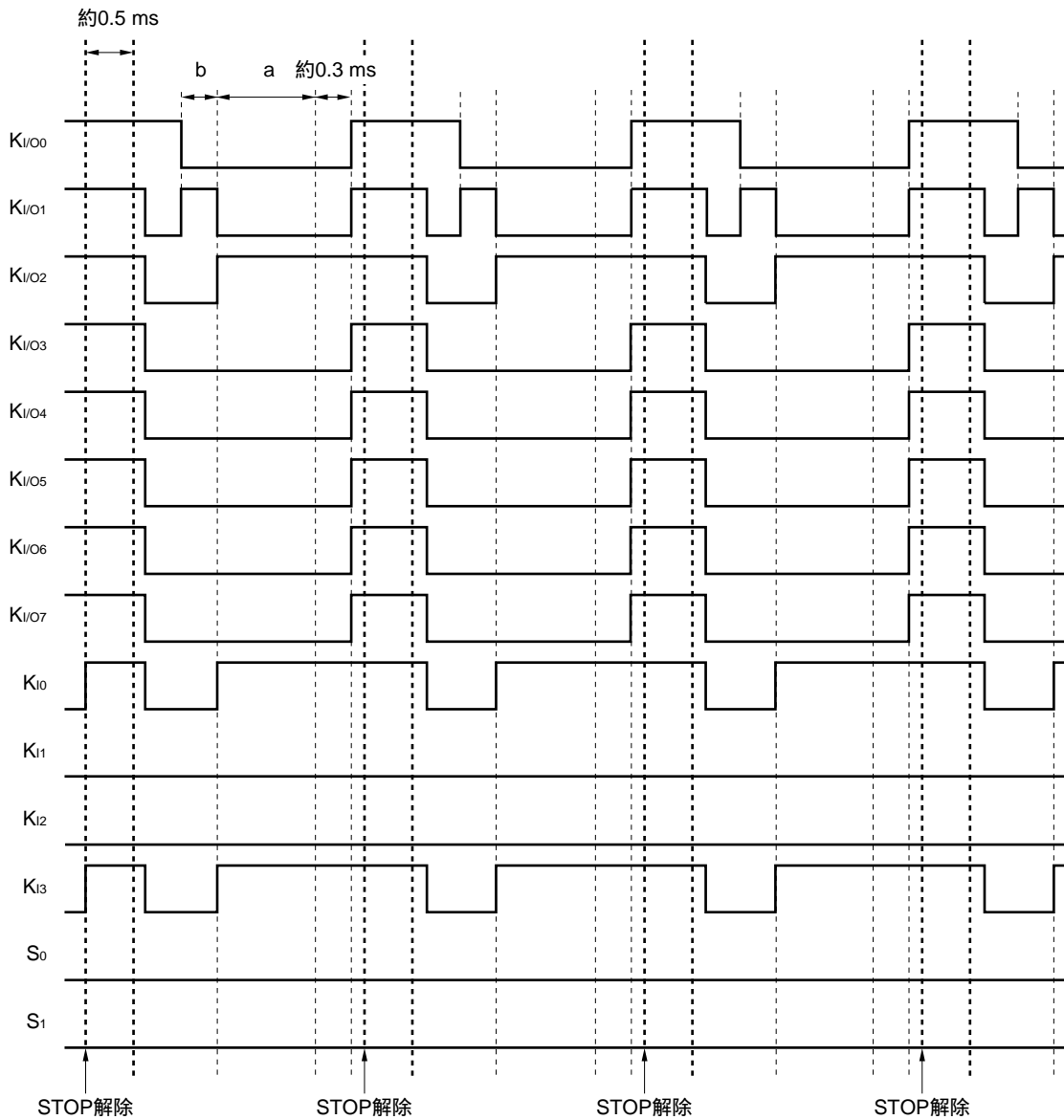
実行時間は約0.41 msです。

c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。

実行時間は約4.23 ms ~ 4.84 msかかります。

図3 - 3 キー確定までの出力パターン (4/5)

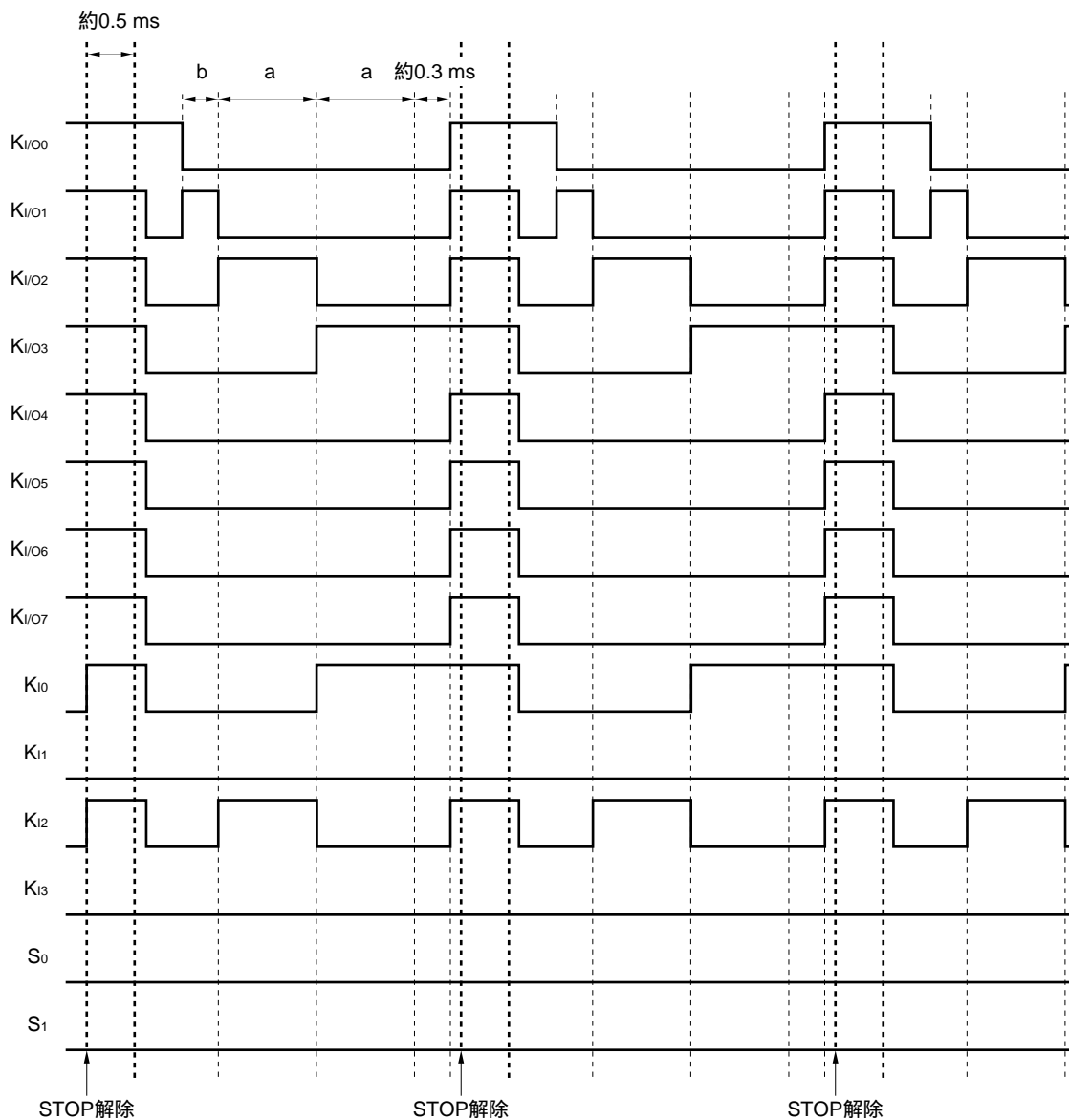
(4) K9とK12が押され、同一キー・ソース間で無効となった場合



- a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
実行時間は約1.34 ms ~ 1.95 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。

図3 - 3 キー確定までの出力パターン (5/5)

(5) K10とK16が押され、異なったキー・ソース間で無効となった場合



a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。

実行時間は約1.34 ms ~ 1.95 msです。

b : キー入力のチェックを行います。

実行時間は約0.41 msです。

3.2.2 二重押しキーの動作

二重押しキーはK29キーです。K29キーとK30キー，K29キーとK31キー，K29キーとK32キーの3種類の二重押しが有効です。

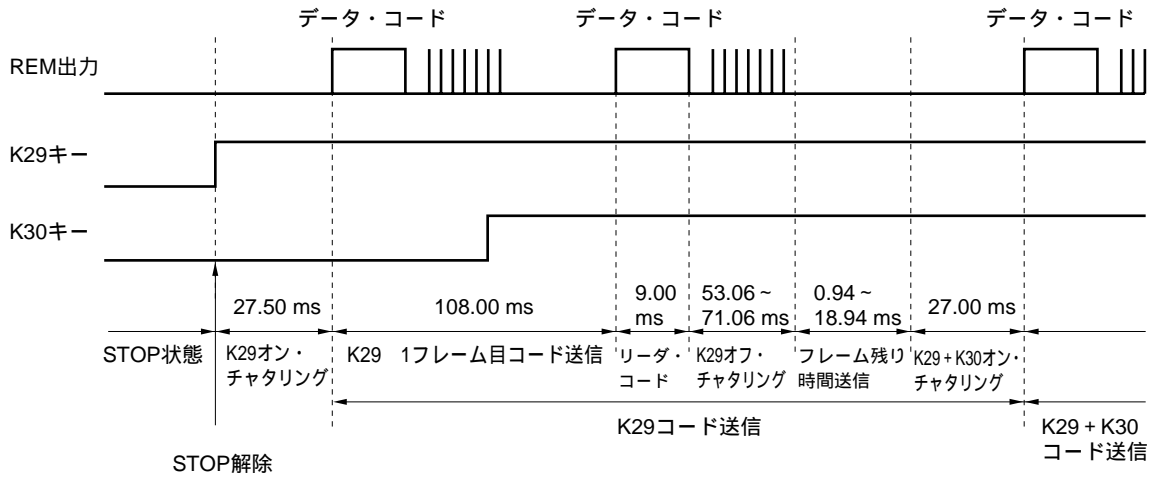
どのパターンもK29キーが先に押されている状態でのみ有効になります。

すなわち，K29以外のキーを先に押すか，K29キーと同時に二重押しを行った場合には，無効キーとなります。

二重押し時のキー動作例を，図3 - 4に示します。

図3 - 4 二重押しキー動作のタイミング・チャート (1/2)

(1) K29の1フレーム送信中に，二重押しの可能なキー (K30-K32) を押した場合，K29コードの2フレーム目を送信したあと，二重押しが有効となり，二重押しのコードを送信します。



(2) K29の3フレーム目送信中に，二重押しの可能なキー (K30-K32) を押した場合，K29コードと3フレーム目を送信したあと，二重押しが有効となり，二重押しのコードを送信します。

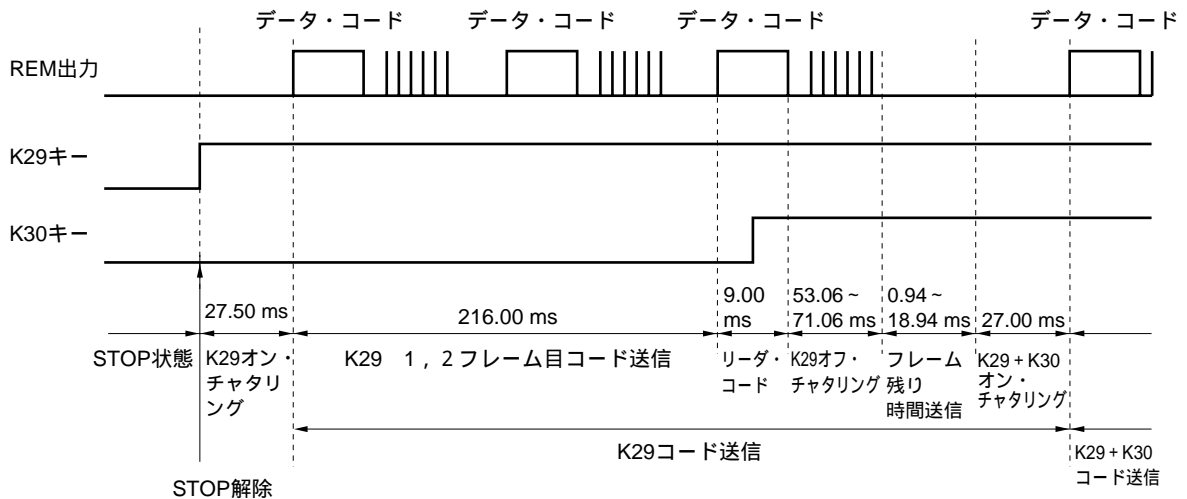
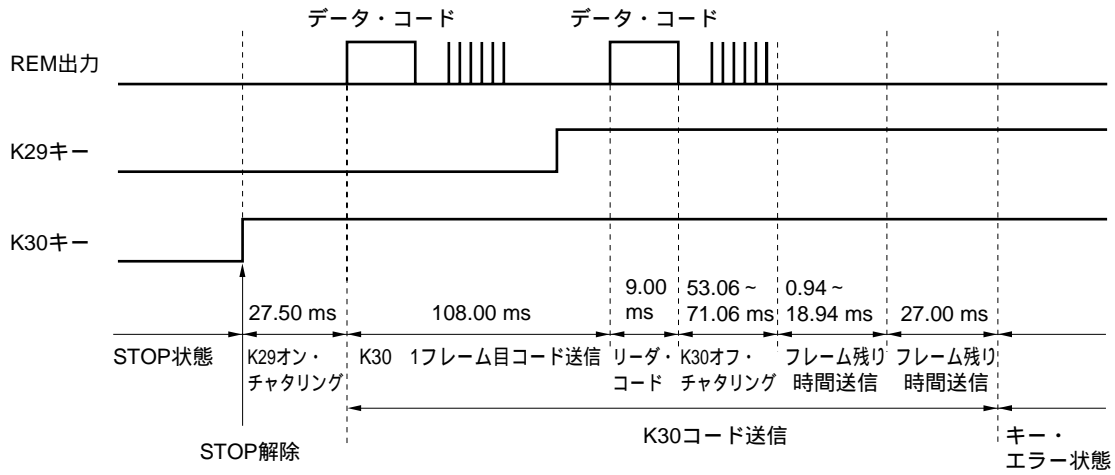
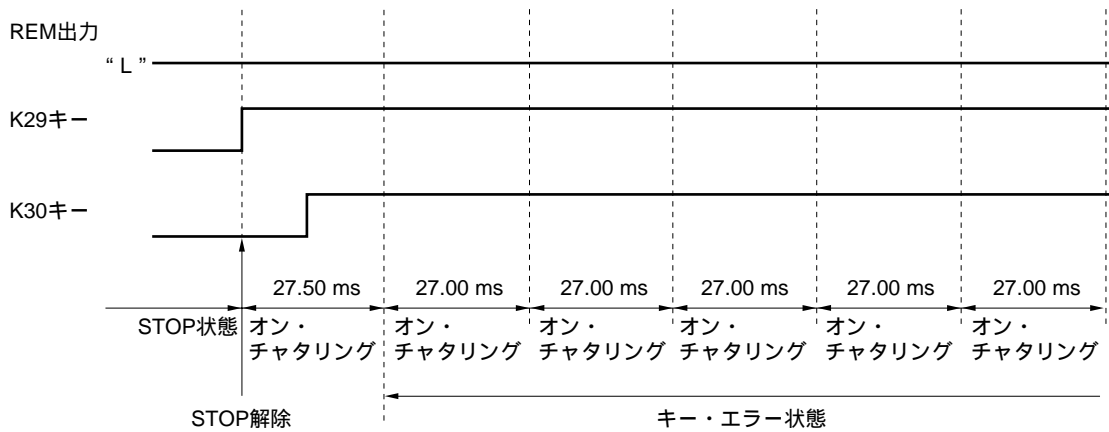


図3-4 二重押しキー動作のタイミング・チャート(2/2)

(3) 二重押しの可能なキー (K30-K32) が確定したあとにK29を押した場合、キー・チェックでキー・エラーとなり、コード送信は行われません (二重押しは無効)。



(4) K29が確定する前に二重押しの可能なキー (K30-K32) を押した場合、キー・チェックでキー・エラーとなり、コード送信は行われません (二重押しは無効)。



3.2.3 キーの渡り操作

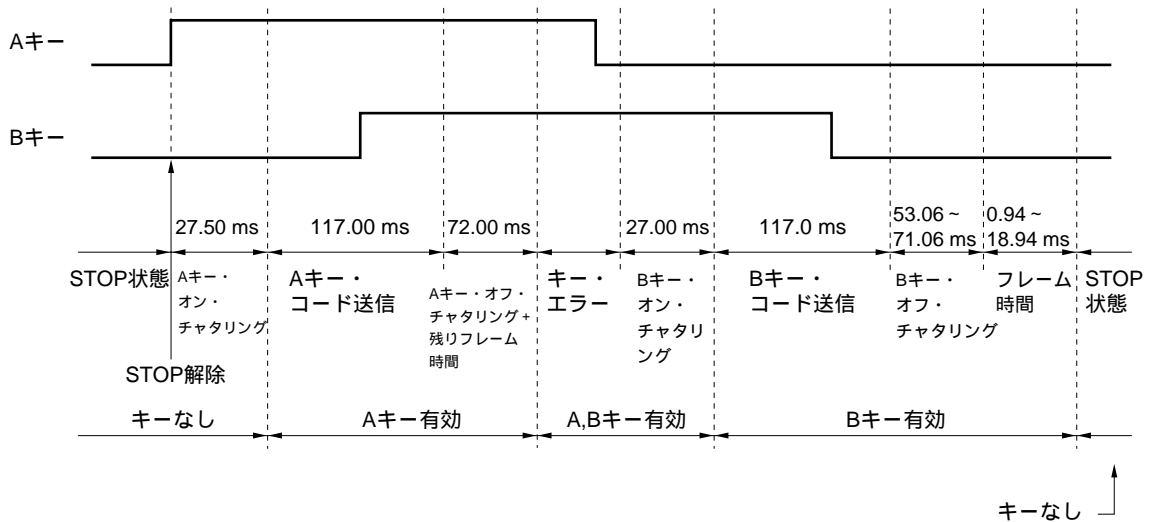
二重押しが無効なキーは、キーの渡り操作ができます。

キーの渡り操作とは、あるキーが離される前に、別のキーが押されるような操作のことです。一方のキーが離された時点で、押し続けられている他方のキーが有効となります。

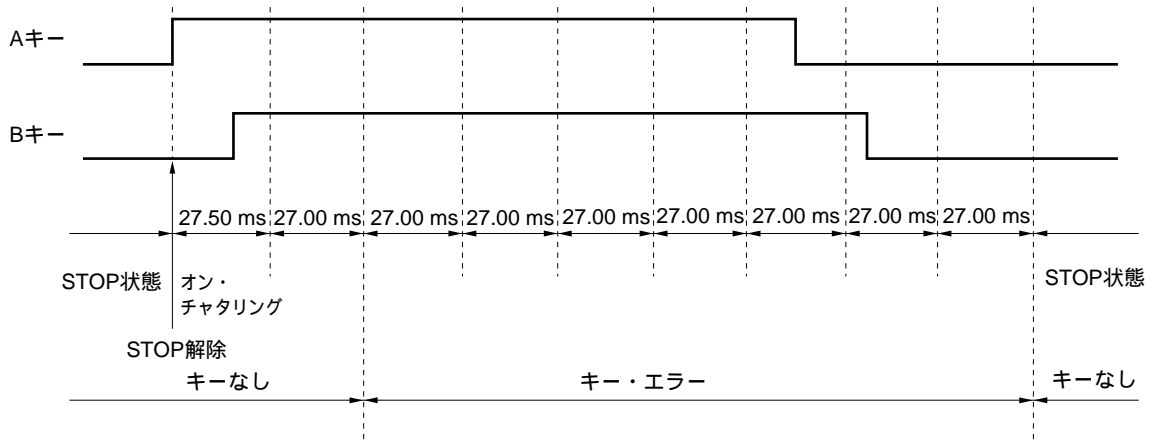
キーの渡り操作時の動作タイミング例を図3 - 5 に示します。

図3 - 5 キーの渡り操作時の動作タイミング例

- (1) Aキーのコード送信中にBキーとの渡り操作が行われた場合、送信途中のコードをすべて送信したあと、キー・チェックを行います。二重押し期間はキー・エラーとなります。
Aキーが離されるとBキーが有効となりコード送信が行われます。



- (2) Aキーが確定する前にBキーとの渡り操作が行われた場合、キー・チェックでキー・エラーとなります。また、二重押し期間もキー・エラーとなります。
Aキーが離されたあと、Bキーのキー・チェックを行います。Bキーはキーありと判断される前に離されているため、キーなしと判断されます。



(メ モ)

第4章 出力コード

出力コードを表4 - 1 に示します。

NECフォーマットで出力する場合、当社にてカスタム・コードをお客様ごとに割り当てることによって、NECフォーマットを出力する別のリモコンからの干渉を防いでいます。

このプログラムでは、カスタム・コードを0AH、カスタム・コード'をF5Hとしています。

カスタム・コードの取得申請については、当社販売員までご相談ください。

表4 - 1 出力コード

キーNo.	カスタム・コード	カスタム・コード'	データ	キーNo.	カスタム・コード	カスタム・コード'	データ	キーNo.	カスタム・コード	カスタム・コード'	データ
K1	0AH	F5H	00H	K17	0AH	F5H	50H	K33	0AH	F5H	A0H
K2	0AH	F5H	01H	K18	0AH	F5H	51H	K34	0AH	F5H	A1H
K3	0AH	F5H	02H	K19	0AH	F5H	52H	K35	0AH	F5H	A2H
K4	0AH	F5H	03H	K20	0AH	F5H	53H	K36	0AH	F5H	A3H
K5	0AH	F5H	04H	K21	0AH	F5H	54H	K37	0AH	F5H	A4H
K6	0AH	F5H	05H	K22	0AH	F5H	55H	K38	0AH	F5H	A5H
K7	0AH	F5H	06H	K23	0AH	F5H	56H	K39	0AH	F5H	A6H
K8	0AH	F5H	07H	K24	0AH	F5H	57H	K40	0AH	F5H	A7H
K9	0AH	F5H	08H	K25	0AH	F5H	58H	K41	0AH	F5H	A8H
K10	0AH	F5H	09H	K26	0AH	F5H	59H	K42	0AH	F5H	A9H
K11	0AH	F5H	0AH	K27	0AH	F5H	5AH	K43	0AH	F5H	AAH
K12	0AH	F5H	0BH	K28	0AH	F5H	5BH	K44	0AH	F5H	ABH
K13	0AH	F5H	0CH	K29	0AH	F5H	5CH	K45	0AH	F5H	ACH
K14	0AH	F5H	0DH	K30	0AH	F5H	5DH	K46	0AH	F5H	ADH
K15	0AH	F5H	0EH	K31	0AH	F5H	5EH	K47	0AH	F5H	AEH
K16	0AH	F5H	0FH	K32	0AH	F5H	5FH	K48	0AH	F5H	AFH

キーNo.	カスタム・コード	カスタム・コード'	データ
K29 + K30	0AH	F5H	90H
K29 + K31	0AH	F5H	91H
K29 + K32	0AH	F5H	92H

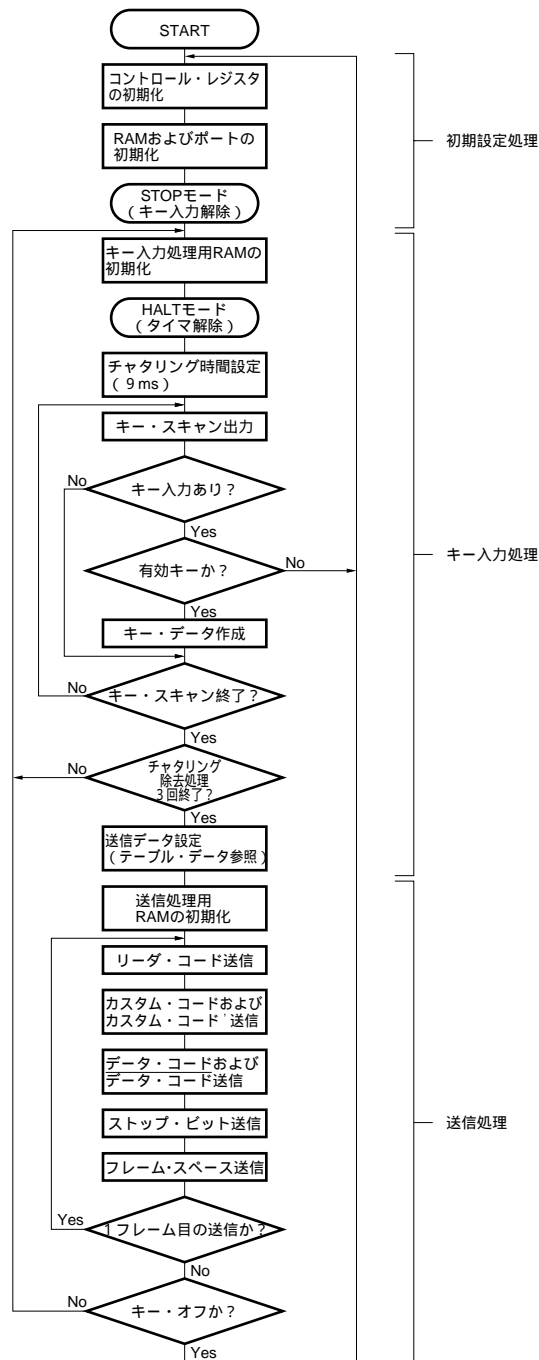
(メ モ)

第5章 ソフトウェア構成

5.1 ジェネラル・フロー・チャート

このプログラムのジェネラル・フロー・チャートを図5-1に示します。

図5-1 ジェネラル・フロー・チャート



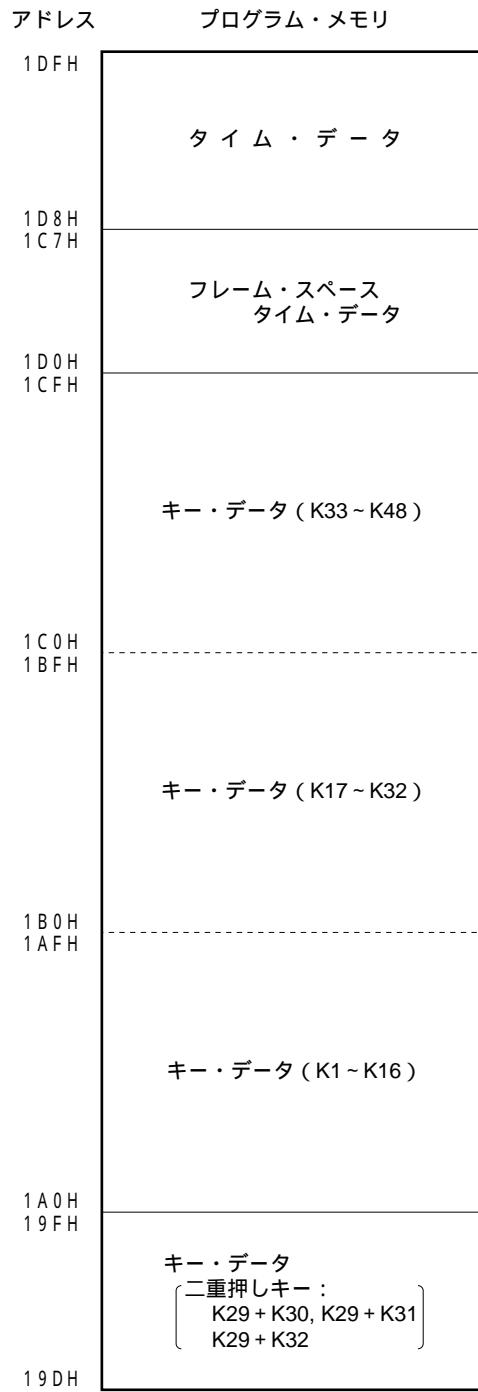
5.2 プログラム・メモリ (ROM) 構成

μPD6133のプログラム・メモリ (ROM) は、512ステップ×10ビット構成になっています。
このプログラムのROMマップを図5 - 2に、データ・テーブル・マップを図5 - 3に示します。

図5 - 2 ROMマップ

アドレス	プログラム・メモリ
1FFH	空 き 領 域
1ECH 1EBH	
1E0H 1DFH	フ レーム・ス ペース出力 サ ブ ル ー チ ン
19DH 19CH	デ ー タ・テ ー ブ ル (図 5 - 3 参 照)
19AH 199H	空 き 領 域
0F4H 0F3H	送 信 処 理
013H 012H	キ ー 入 力 処 理
000H	初 期 設 定

図5 - 3 データ・テーブル・マップ



5.3 データ・メモリ (RAM) 構成

データ・メモリ (RAM) は32×4ビット構成のスタティックRAMで、処理データのストアに利用します。命令によっては8ビット単位で操作することもできます。

R0はROMアドレスのデータ・ポインタとしての機能を持っています。このデータ・ポインタにROMアドレスを設定し、ROMの内容を読み出すことができます。これをROMデータのテーブル参照と呼びます。リセット時には、R0は00Hになります。

また、RFはアドレス・スタック・レジスタと兼用になっています。リセット時にはR1～RFは不定となります。

プログラム全体 (キー入力処理 + 送信処理)、キー入力処理、送信処理のRAMマップを表5 - 1から表5 - 3に示します。また、RAM使用説明を表5 - 4に示します。

表5 - 1 RAMマップ

	H (1)	L (0)
R0	ワーク1 データ・ポインタH	チャタリング・カウンタ データ・ポインタL
R1	確定キー・データH	確定キー・データL
R2	イミューディエト (0EH) 設定用	イミューディエト (1H) 設定用
R3	K29 ONフラグ	連続フラグ
R4	キー・データH カスタム・コードH	キー・データL カスタム・コードL
R5	カスタム・コード ' H	カスタム・コード ' L
R6	キーありフラグ データ・コードH	キー・スキャン・カウンタ データ・コードL
R7	イミューディエト (0FH) 設定用	キー・オフ・チェック・カウンタ
R8	キー・スキャン・データH イミューディエト (0FH) 設定用 データ1	キー・スキャン・データL イミューディエト (0H) 設定用 データ0
R9	比較キー・データH データ1送信カウンタH	比較キー・データL データ1送信カウンタL
RA	ワーク2	キー・リターン・チェック・カウンタ 送信ビット・カウンタ
RB	イミューディエト (0CH) 設定用	イミューディエト (3H) 設定用
RC	未使用	未使用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	

表5 - 2 キー入力処理時使用RAMマップ

	H (1)	L (0)
R0	ワーク1 データ・ポインタH	チャタリング・カウンタ データ・ポインタL
R1	確定キー・データH	確定キー・データL
R2	イミーディエト (0EH) 設定用	イミーディエト (1H) 設定用
R3	K29 ONフラグ	連続フラグ
R4	キー・データH カスタム・コードH	キー・データL カスタム・コードL
R5	カスタム・コード ' H	カスタム・コード ' L
R6	キーありフラグ データ・コードH	キー・スキャン・カウンタ データ・コードL
R7		
R8	キー・スキャン・データH イミーディエト (0FH) 設定用	キー・スキャン・データL イミーディエト (0H) 設定用
R9	比較キー・データH	比較キー・データL
RA		キー・リターン・チェック・カウンタ
RB	イミーディエト (0CH) 設定用	イミーディエト (3H) 設定用
RC	未使用	未使用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	

: 送信処理で使用

表5 - 3 送信処理時使用RAMマップ

	H (1)	L (0)
R0	データ・ポインタH ワーク1	データ・ポインタL
R1		
R2		
R3		連続フラグ
R4	カスタム・コードH	カスタム・コードL
R5	カスタム・コード' H	カスタム・コード' L
R6	データ・コードH	データ・コードL
R7	イミューディエト (OFH) 設定用	キー・オフ・チェック・カウンタ
R8	データ1	データ0
R9	データ1送信カウンタH	データ1送信カウンタL
RA	ワーク2	送信ビット・カウンタ
RB		
RC	未使用	未使用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	


 : キー入力処理で使用

表5 - 4 RAM使用説明(1/3)

名 称	RAM	説 明						
データ・ポインタ	R0	ROMアドレスを指定するためのポインタです。 ROMデータのテーブル参照を行う際に使用します。						
チャタリング・カウンタ	R00	キー取り込み時のチャタリング回数をカウントするためのカウンタです。 3回カウントするためイニシャル値として、0CHが設定されます。0FHで3回終了と判断します。						
ワーク1	R10	ワーク・エリアです。 データを一時的に格納するためのエリアです。						
確定キー・データ	R1	オン・チャタリング除去処理終了後に、確定したキー・データを格納します。 すでに確定キー・データが格納されている場合には、チャタリング除去処理終了時のキー・データ(R4)と比較して一致すれば、このデータは保持されます。不一致の場合には、新たにキー・データ(R4)の値が格納されず。						
イミューディエト(1H)設定用	R02	上位3ビット・クリアまたは1H設定用に使用します。						
イミューディエト(0EH)設定用	R12	最下位ビット・クリアまたは0EH設定用に使用します。						
連続フラグ	R03	送信処理で、現在1フレーム目を送信中か、2フレーム目以降を送信中かを判断するためのフラグです。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>2フレーム目以降を送信中</td> </tr> <tr> <td>0H</td> <td>1フレーム目を送信中</td> </tr> </tbody> </table> 0FH：セット，0H：クリア	値	説 明	0FH	2フレーム目以降を送信中	0H	1フレーム目を送信中
値	説 明							
0FH	2フレーム目以降を送信中							
0H	1フレーム目を送信中							
K29 ONフラグ	R13	K29キー(二重押しが可能なキー)が押されたかどうかを判断するためのフラグです。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>K29キーが押された場合</td> </tr> <tr> <td>0H</td> <td>K29以外のキーが押された場合</td> </tr> </tbody> </table> 0FH：セット，0H：クリア	値	説 明	0FH	K29キーが押された場合	0H	K29以外のキーが押された場合
値	説 明							
0FH	K29キーが押された場合							
0H	K29以外のキーが押された場合							
キー・データ	R4	オン・チャタリング中での、前回のキー・データを格納します。 比較キー・データ(R9)の値と比較して一致すれば、このデータは保持されます。不一致の場合または初回には、新たに比較キー・データ(R9)の値が格納されます。						
カスタム・コード	R4	カスタム・コードを格納します。						
カスタム・コード'	R5	カスタム・コード'を格納します。						
データ・コード	R6	データ・コードを格納します。						
キー・スキャン・カウンタ	R06	キー・スキャン回数をカウントするためのカウンタです。 8回カウントする必要があるため、イニシャル値として8H(8回)が設定されます。						

表5 - 4 RAM使用説明 (2/3)

名 称	RAM	説 明										
キーありフラグ	R16	キーが押されているかどうかを判断するためのフラグです。										
		<table border="1"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>キーが押されている場合</td> </tr> <tr> <td>0H</td> <td>キーが離されている場合</td> </tr> </tbody> </table>	値	説 明	0FH	キーが押されている場合	0H	キーが離されている場合				
		値	説 明									
		0FH	キーが押されている場合									
0H	キーが離されている場合											
0FH：セット，0H：クリア												
キー・オフ・チェック・カウンタ	R07	<p>ビット・データ送信中のキー・オフの状態の回数をカウントするためのカウンタです。</p> <p>10回カウントする必要があるため、イニシャル値として5Hが設定され、0FH（10回連続キー・オフ状態）になった時点でキー・オフ状態が確定します。キー・オフではないと判断された場合は、その時点でカウンタがイニシャライズされます。</p>										
イミディエト（0FH）設定用	R17	各フラグをセットするために使用します。										
キー・スキャン・データ	R8	キー・スキャン出力データを格納します。										
データ0 データ1	R08 R18	<p>ビット・データ‘0’および‘1’を送信するための時間データ（テーブル・データ）の口ウ・アドレスを格納します。</p> <p>データ0：ビット・データ‘0’ データ1：ビット・データ‘1’</p>										
イミディエト（0H）設定用	R08	各フラグをセットするために使用します。										
イミディエト（0FH）設定用	R18	<p>フラグ・セットおよび、データ反転用に使用します。</p> <p>設定値は0FHです。</p>										
比較キー・データ	R9	オン・チャタリング中のキー・データを格納します。										
データ1送信カウンタ	R9	送信したビット・データ‘1’の数をカウントするためのカウンタです。										
キー・リターン・チェック・カウンタ	R0A	<p>キー・リターン・データにより、キー・データを算出するためのカウンタです。</p> <p>K_i入力の場合は4回、S₀またはS₁入力の場合は2回カウントする必要があるため、イニシャル値は0CH（4回）または0EH（2回）を設定します。</p>										
送信ビット・カウンタ	R0A	ビット送信処理を行う際、ワーク2（R1A）に格納されたビット・データを、必要なビット数送信するためのカウンタです。1～4回のカウンタです。										
		<table border="1"> <thead> <tr> <th>設定値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>1ビット送信（リーダ・コード，ストップ・ビット送信時）</td> </tr> <tr> <td>0EH</td> <td>2ビット送信（未使用）</td> </tr> <tr> <td>0DH</td> <td>3ビット送信（未使用）</td> </tr> <tr> <td>0CH</td> <td>4ビット送信（カスタム・コード，カスタム・コード’，データ・コード，$\overline{\text{データ・コード}}$送信時）</td> </tr> </tbody> </table>	設定値	説 明	0FH	1ビット送信（リーダ・コード，ストップ・ビット送信時）	0EH	2ビット送信（未使用）	0DH	3ビット送信（未使用）	0CH	4ビット送信（カスタム・コード，カスタム・コード’，データ・コード， $\overline{\text{データ・コード}}$ 送信時）
		設定値	説 明									
		0FH	1ビット送信（リーダ・コード，ストップ・ビット送信時）									
		0EH	2ビット送信（未使用）									
0DH	3ビット送信（未使用）											
0CH	4ビット送信（カスタム・コード，カスタム・コード’，データ・コード， $\overline{\text{データ・コード}}$ 送信時）											

表5 - 4 RAM使用説明(3/3)

名 称	RAM	説 明
ワーク2	R1A	ビット送信を行うための出力データを格納します。
イミューティエト(3H)設定用	R0B	データ判断, 3H設定用に使用します。
イミューティエト(0CH)設定用	R1B	データ判断, 0CH設定用に使用します。

5.4 フラグ・マップ

各処理中でのフラグの動きを、表5 - 5に示します。

表5 - 5 フラグ・マップ

(1) 連続フラグ (R03)

処 理		R03
初期設定処理		クリア (0H)
キー入力処理	オン・チャタリング除去処理終了時のキー・データが、前に行ったオン・チャタリング除去処理時の確定キー・データと異なる場合	クリア (0H)
送信処理	毎フレームのフレーム・スペース送信中	判断
	1フレーム目のみ、フレーム・スペース送信中	セット (0FH)

判断：その処理中に判断を行っている

(2) K29 ONフラグ (R13)

処 理		R13
初期設定処理		クリア (0H)
キー入力処理	キーありフラグがセットされている場合	判断
	チャタリングが3回終了し、確定キー・データがK29であった場合	セット (0FH)
	チャタリングが3回終了し、確定キー・データがK29、有効二重押しのどちらでもない場合	クリア (0H)
送信処理		-

判断：その処理中に判断を行っている

- : 使用していない

(3) キーありフラグ (R16)

処 理		R16
初期設定処理		-
キー入力処理	RAMの初期化時	クリア (0H)
	キー・リターン・チェックの冒頭	判断
	キー・データ算出後	判断
	キー・リターン・チェック中、キー入力があった場合	セット (0FH)
	キー・スキャン (8回分) 終了後	判断
送信処理		×

判断：その処理中に判断を行っている

- : 使用していない × : 別の用途で使用している

第6章 プログラム説明

6.1 初期値設定

赤外線リモート・コントロール送信機用マイクロコントローラは、通常、電源として電池を使用しています。また赤外線LEDは、発光時に大きな電流を消費します。そのため、赤外線LEDの発光時に電源電圧が大きく変動することがあり、これによってRAMやポートの内容が突然変化する場合があることを想定しておく必要があります。

RAMやポートの突然の変化による誤動作を防ぐため、送信を一回行うごとに初期値を設定し直すようにプログラムを設計します。

6.1.1 処理説明

(1) ポートの設定およびコントロール・レジスタの初期値を設定します。

K_{I/O}ポート (P0)

キー・スキャン出力用の8ビットの入出力ポートです。

初期設定値はFFHです。すべてのビット (K_{I/O0}-K_{I/O7}) はハイ・レベル出力に設定されます。

コントロール・レジスタ0 (P3)

コントロール・レジスタ0の内容を表6-1と表6-2に示します。

初期設定値は13Hです。表中の網掛けのように設定されます。

表6-1 コントロール・レジスタ0 (P3)

ビット		b7	b6	b5	b4	b3	b2	b1	b0
名称		-	-	DP(データ・ポインタ)		TCTL	CARY	MOD ₁	MOD ₀
				DP ₉	DP ₈				
設定値	0	“ 0 ”固定	“ 0 ”固定	0	0	1/1	ON	表6-2参照	
	1			1	1	1/2	OFF		
リセット時		0	0	0	0	0	0	1	1

b₀, b₁..... REM出力のキャリア周波数とDuty比を指定します。

b₂..... b₀, b₁で指定される周波数のキャリアの有無を指定します。

“ 0 ” = ON (キャリアあり), “ 1 ” = OFF (キャリアなし, ハイ・レベル)。

b₃..... キャリア周波数とタイマ・クロックの分周比を変更します。

“ 0 ” = 1/1 (キャリア周波数: b₀, b₁の指定値, タイマ・クロック: f_x/8)。

“ 1 ” = 1/2 (キャリア周波数: b₀, b₁の指定値の1/2, タイマ・クロック: f_x/16)。

表6-2 タイマ・クロックとキャリア周波数の設定

b ₃	b ₂	b ₁	b ₀	タイマ・クロック	キャリア周波数 (Duty比)
0	0	0	0	f _x /8	f _x (Duty 1/2)
		0	1		f _x /8 (Duty 1/2)
		1	0		f _x /12 (Duty 1/2)
		1	1		f _x /12 (Duty 1/3)
	1	x	x	キャリアなし (ハイ・レベル)	
1	0	0	0	f _x /16	f _x /2 (Duty 1/2)
		0	1		f _x /16 (Duty 1/2)
		1	0		f _x /24 (Duty 1/2)
		1	1		f _x /24 (Duty 1/3)
	1	x	x	キャリアなし (ハイ・レベル)	

b₄, b₅.....ROMのデータ・ポインタの上位2ビット (DP₈, DP₉) の指定を行います。

備考1 . x : don't care

2 . : 初期設定値 (13H)

3 . f_x : システム・クロック周波数

コントロール・レジスタ1 (P4)

コントロール・レジスタ1の内容を表6-3に示します。

初期設定値は33Hです。表中の網掛けのように設定されます。

表6-3 コントロール・レジスタ1 (P4)

ビット		b7	b6	b5	b4	b3	b2	b1	b0
名称		-	-	Ki pull-down	So/S1 pull-down	-	S1/ $\overline{\text{LED}}$ mode	Ki/o mode	So mode
設定値	0	" 0 " 固定	" 0 " 固定	OFF	OFF	" 0 " 固定	S1	IN	OFF
	1			ON	ON		$\overline{\text{LED}}$	OUT	IN
リセット時		0	0	1	0	0	1	1	0

b0.....Soポートの入力モードを指定します。“ 0 ” = OFFモード (ハイ・インピーダンス) , “ 1 ” = IN (入力モード)。

b1.....Ki/oポートの入出力モードを指定します。“ 0 ” = IN (入力モード) , “ 1 ” = OUT (出力モード)。

b2.....S1/ $\overline{\text{LED}}$ ポートの入出力モードを指定します。“ 0 ” = S1 (入力モード) , “ 1 ” = $\overline{\text{LED}}$ (出力モード)。

b4.....So/S1ポート入力モード時のプルダウン抵抗の有無を指定します。“ 0 ” = OFF (なし) , “ 1 ” = ON (あり)。

b5.....Kiポート入力モード時のプルダウン抵抗の有無を指定します。“ 0 ” = OFF (なし) , “ 1 ” = ON (あり)。

備考1 . 出力モード, OFFモード時は, プルダウン抵抗はすべて自動的に切れます。

2 . : 初期設定値 (33H)

(2) RAMの初期値を設定します。

次に示すRAMが0にクリアされます。

- ・ 確定キー・データ (R1)
- ・ 連続フラグ (R03)
- ・ K29 ONフラグ (R13)

(3) STOPモードを設定します。

HALT命令の解除条件一覧表を表6-4に示します。

初期設定値は8Hです。初期設定時(STOPモード)の解除条件は、表中の網掛けのように設定されます。

表6-4 HALT命令の解除条件一覧表

HALT命令のオペランド値				設定モード	設定のための前提条件	解除条件
b ₃	b ₂	b ₁	b ₀			
0	0	0	0	STOP	K _{I/O} 端子がすべてハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
	0	1	1	STOP	K _{I/O} 端子がすべてハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
	1	1	0	STOP ^{注1}	K _{I/O0} 端子がハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
1	上記のb ₂ b ₁ b ₀ の組み合わせのうちいずれか			STOP	【上記のほかに次の条件が追加される】 ----- -	S ₀ , S ₁ 端子の少なくとも1つにハイ・レベルを入力 ^{注2}
0/1	1	0	1	HALT	-	タイマのダウン・カウンタが0になったとき

注1 . HALT #x110Bを設定する場合には、暴走時に内部リセットがかかるようにK_{I/O0}端子とK_I端子を使ってキー・マトリクスを構成してください。

2 . S₀, S₁端子の少なくともどちらか1端子(スタンバイ解除に使用する端子)が、入力モードでなければなりません(ただし、両方とも出力モードの場合であっても内部リセットはかかりません)。

注意1 . 上記以外のオペランド値でHALT命令が実行された場合や、HALT命令の実行時に前提条件が満足されていない場合には、内部リセットがかかります。

2 . タイマのダウン・カウンタが0でないとき(タイマ動作中)にSTOPモードを設定すると、タイマのダウン・カウンタとタイマ出力の許可フラグの全10ビットが0にクリアされてから、STOPモードになります。

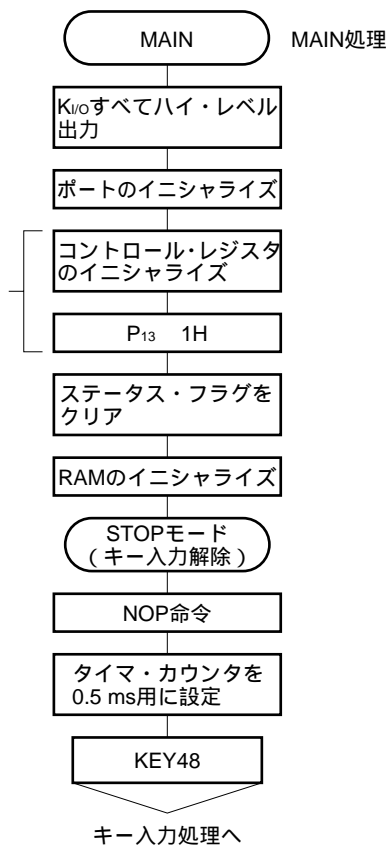
3 . STOPモード解除後の最初の命令はNOP命令を記述してください。

備考 : 初期設定値(8H)

(4) STOPモード解除後、NOP命令を実行してから、タイマに初期値を設定します。

タイマ初期設定値は1CH(=0.5ms)です。

6.1.2 ディテール・フロー・チャート



☆☆☆☆☆☆ 初期設定処理 ☆☆☆☆☆☆

```

TIME05M      EQU      01CH          :0.5ms
DamyTime     EQU      512-1        :9.00ms

;##### P U B L I C #####
PUBLIC       MAIN
;
;##### E X T E R N #####
;

;##### S T A R T #####
;*****
; Control Register Initialize
;*****
MAIN:
OUT          P0,#0FFH              ① キー・スキャン出力 (K1/0) をすべてハイ・レベルに設定します。
OUT          P4,#033H              ② ポートのインシャライズを行います。
                                       S0, S1:入力モード, K1/0:出力モード
OUT          P3,#013H              ③ コントロール・レジスタ0, 1 (P3, P4) のインシャライズをします。
                                       キャリアあり, 周波数:fx/12, デューティ:1/3,
                                       TCTL:1/1, Ki:プルダウン抵抗あり,
                                       S0, S1:プルダウン抵抗あり
                                       データ・ポインタ:DP9, DP9=01H (テーブル参照用アドレスの最上位桁)

MOV          T,#DamyTime           }----- ④ ステータス・フラグをクリアします。
STTS        #0101B

;*****
; RAM Initialize Routine
;*****
MOV          R1,#000H
MOV          R3,#000H
HALT        #008H                  ⑥ STOPモード:KiおよびS0, S1ハイ・レベル入力により解除されます。
NOP
MOV          T,#TIME05M            ⑦ NOP命令
;
;                               ⑧ タイマ値のインシャライズを行います。
;                               1CH=0.5ms
END

```

6.2 キー入力処理

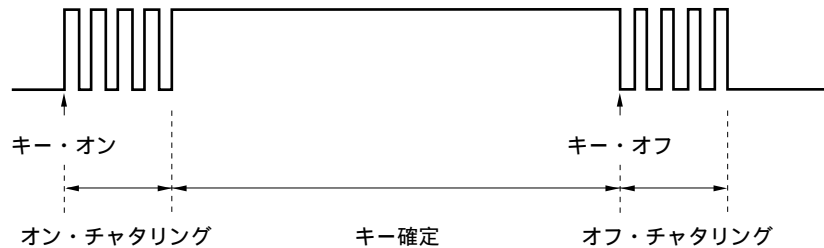
6.2.1 処理説明

キー入力処理はチャタリング除去処理，キー・スキャン処理，カスタム・コード作成処理，キー・データ作成処理により構成されています。

(1) チャタリング除去処理

キーをオンまたはオフにする場合，キー信号がハイ・レベルまたはロウ・レベルに安定するまでの間に，チャタリング（オン・チャタリング，オフ・チャタリング）と呼ばれる不安定な状態が存在します（図6-1を参照）。キー入力は，このチャタリング中では不安定な状態であるため，プログラムでチャタリングを除去する必要があります。

図6-1 キー入力信号のチャタリング



次にチャタリング除去の例（キー・オン判定の例）を図6-2に示します。

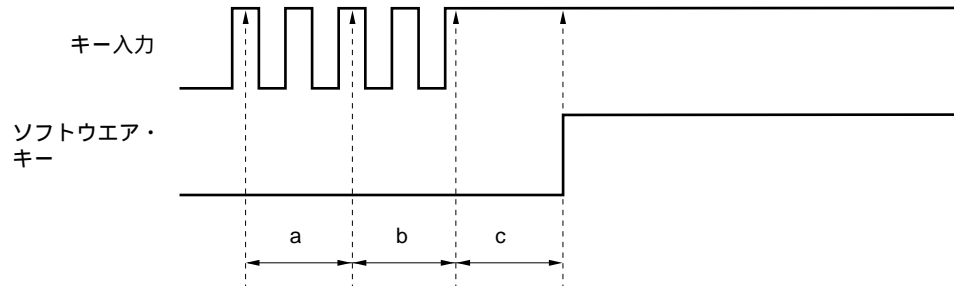
でキーがオンされたことをSTOPモード解除によって認識し，その後一定間隔（ \sim : 9.00 ms \times 3回）で，キー入力をチェックします。

(a)では，チェック位置ですべてキー・オンされているので，ソフトウェア・キーの の位置でキー信号がハイ・レベルになったと判断します。

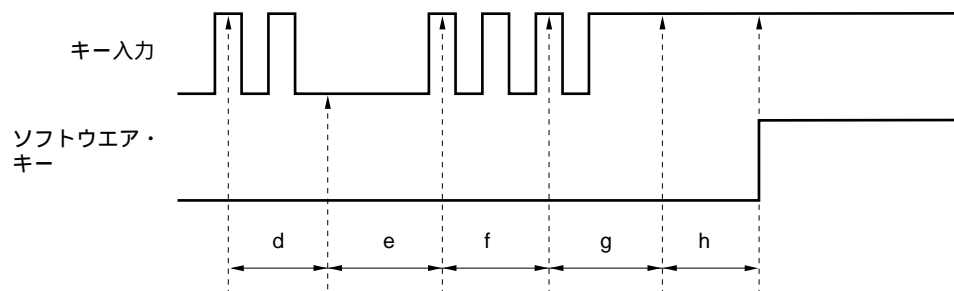
でキーがオンされたと認識してから，ソフトウェア上でハイ・レベルになったと判断するまでの待ち時間（ $a + b + c$ ）をチャタリング除去時間といいます。（a）では，キー入力をチェックするときに常にオンされていたので，チャタリング除去時間は $a + b + c$ になります。（b）では，チャタリング除去時間は $d + e + f + g + h$ になります。

図6-2 チャタリング発生時のキー・オン判定

(a)



(b)



(2) キー・スキャン処理

キー・スキャン処理を , に示します。

キー・マトリクス

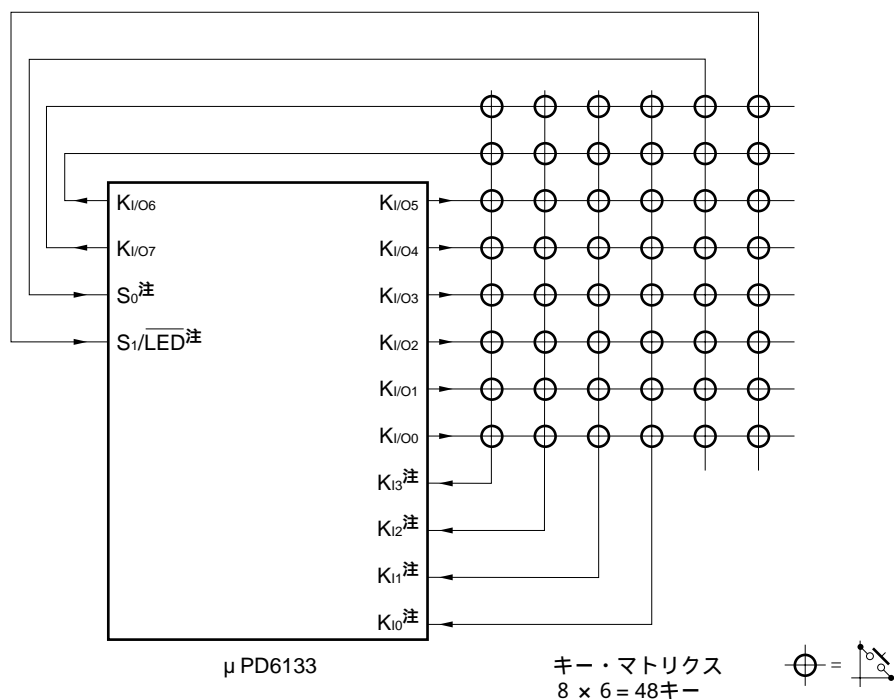
図6 - 3 に示す48個のキー・マトリクスを例にとって説明します。

この例では, K_{I/O0}-K_{I/O7}は出力ポートとしてキー・スキャン信号を出力します。

この信号をK₁₀-K₁₃, S₀, S₁より入力ポートとして取り込みます(キー・リターン信号)。

キー・リターン信号入力ポート(K₁₀-K₁₃, S₀, S₁)には, 内蔵プルダウン抵抗がプログラムによって接続されているので, キーが押されていないときにはロウ・レベルが入力されます。

図6 - 3 48キー・マトリクス



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

キー・スキャン

48キー・マトリクスのうち, どのキーが押されたか判断するには, まず, STTS命令でK₁₀-K₁₃, およびS₀, S₁に入力(ハイ・レベル)があったかを確認します。

次に, キー・スキャン信号出力用ポート(K_{I/O0}-K_{I/O7})をK_{I/O0}から順に1つずつハイ・レベルにしていき, K_{I/O0}-K_{I/O7}のどの出力に対して入力があったかを確認します。

プログラム中のキー・スキャン処理部分では、キー・スキャン・カウンタ（キー・ソース位置）、およびキー・リターン・チェック・カウンタ（キー・リターン位置）により、キー・データ（キー位置）が判断されます。キー・データと2つのカウンタ値の対応を図6 - 5に示します。

データ・メモリ上でのキー・データのビット構成については（4）キー・データのビット構成に示します。

キー・スキャンを行う場合、キーボード上の浮遊容量、配線インピーダンスによりキー・ソースの立ち上がりが遅くなることを考慮する必要があります。このプログラムでは、キー・ソース出力をハイ・レベルで出力してから、100 μs程度（455 kHzの場合6ステップ）待ったあとに、キー入力を取り込むようにしています。

図6 - 4 キー・マトリクス

K _{I/O7}	K29	K30	K31	K32	K47	K48
K _{I/O6}	K25	K26	K27	K28	K45	K46
K _{I/O5}	K21	K22	K23	K24	K43	K44
K _{I/O4}	K17	K18	K19	K20	K41	K42
K _{I/O3}	K13	K14	K15	K16	K39	K40
K _{I/O2}	K9	K10	K11	K12	K37	K38
K _{I/O1}	K5	K6	K7	K8	K35	K36
K _{I/O0}	K1	K2	K3	K4	K33	K34
	K ₁₃	K ₁₂	K ₁₁	K ₁₀	S ₁	S ₀



図6 - 5 キー・データ（キー位置）

キー・スキャン・カウンタ（R06）

F	K29	K30	K31	K32	K47	K48
E	K25	K26	K27	K28	K45	K46
D	K21	K22	K23	K24	K43	K44
C	K17	K18	K19	K20	K41	K42
B	K13	K14	K15	K16	K39	K40
A	K9	K10	K11	K12	K37	K38
9	K5	K6	K7	K8	K35	K36
8	K1	K2	K3	K4	K33	K34

キー・リターン・
チェック・カウンタ（R0A）

C D E F E F

(3) カスタム・コード作成処理

このプログラムでは、カスタム・コードとして0AH，カスタム・コード'としてF5Hを送信するように設定しています。

具体的にはカスタム・コードとしてデータ・メモリR4に50H，カスタム・コード'としてR5にAFHを設定しています。

注意 NECフォーマットでは、LSBから送信するため、ビット列の並びを逆にした値を設定します。

(4) キー・データ作成処理

キー・データ処理を ， に示します。

キー・データのビット構成

キー・データは8ビットで構成されています。各ビットはキー・ソース，キー・リターンの状態を表します。

キー・スキャン・カウンタは、K1/00-K1/07の8種類のため、K10-K13の入力の場合にはキー・データHの最下位ビットとキー・データLの上位2ビットを、S0, S1の入力の場合にはキー・データLの上位3ビットを使用します。また、キー・リターン・チェック・カウンタは、K10-K13の場合には4種類のためキー・データLの下位2ビットを、S0, S1の場合には2種類のためキー・データLの最下位ビットを使用します。

最初にキー入力があるかを判断したときに、S0, S1に入力があった場合にはキー・データHに0CHを設定します。このキー・データHは、図6-6、図6-7のどちらのフォーマットで作成するかを判断するときにも使用します。

また、このキー・データは、 **データ・コード**に示すように、データ・コード算出するときの、テーブル参照用アドレスとして使用します。そのため、図6-6、図6-7に示したようなビット構成でキー・データを作成します。

図6-6 K10-K13に入力があったときのキー・データのビット構成

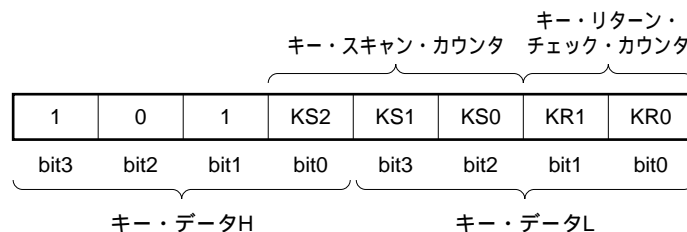
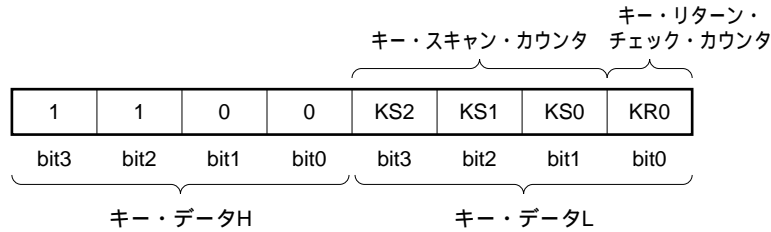


図6-7 S₀, S₁に入力があったときのキー・データのビット構成

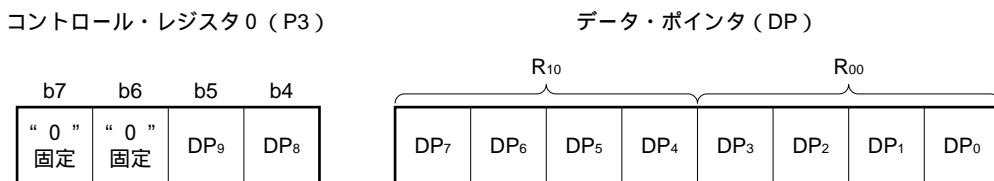


データ・コード

図6-6, 図6-7に示している, キー入力処理によって得られたキー・データを, テーブル参照アドレスとして使用することにより, データ・コードを算出することができます。

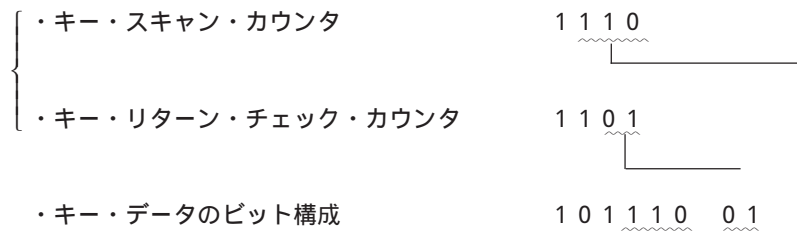
テーブル参照方法は, コントロール・レジスタ0 (P3) の上位4ビットに1Hを設定し, キー・データをデータ・ポインタ (図6-8参照) に設定することにより, ROM内容を送信コードとして使用することができます。

図6-8 データ・ポインタの構成



例 キー位置: K26 (キー入力: K₁₂) の場合

1. キー・スキャン・カウンタ (R06) = EH,
 キー・リターン・チェック・カウンタ (R0A) = DHになります (図6-5参照)。
2. キー・リターン・チェック・カウンタの下位3ビットとキー・リターン・チェック・カウンタの下位2ビットを使用し, 図6-6に示したようなビット構成でキー・データを作成します。

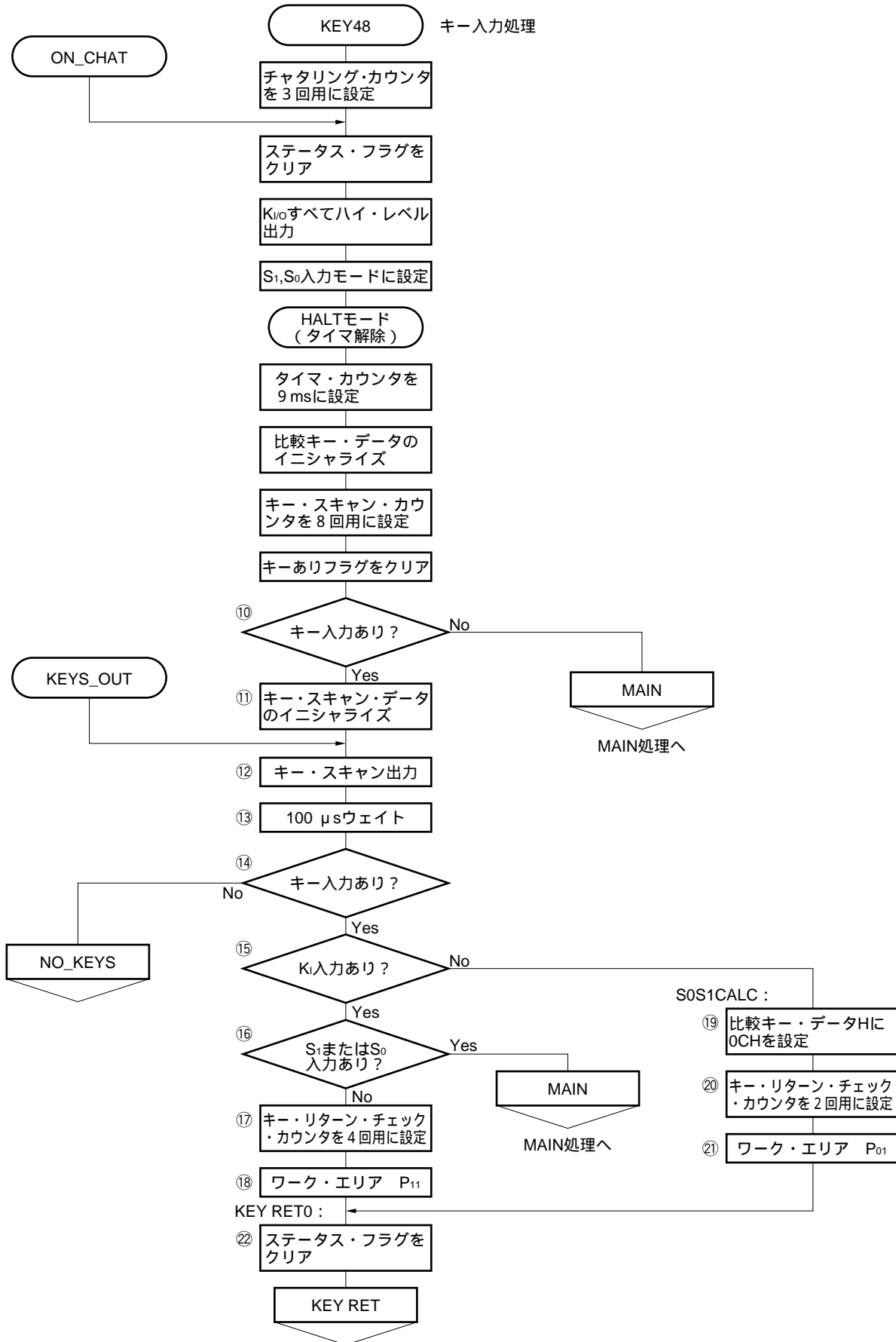


3. キー・データをデータ・ポインタに設定します。
 - ・データ・ポインタの構成 0 0 0 1 1 0 1 1 1 0 0 1 1 B 9 H

図6-6, 図6-7のビット構成でキーを作成すると, 次のようなテーブル参照アドレスになります(なお, 前述のとおり, コントロール・レジスタ0 (P3) の上位4ビットには1Hが設定されています)。

- ・キー入力: K_{10} - K_{13} , キー位置: $K1$ - $K32$
...参照アドレス = 1A0H-1BFH
- ・キー入力: S_0 , S_1 , キー位置: $K33$ - $K48$
...参照アドレス = 1C0H-1CFH
- ・二重押し ($K29 + K30$, $K29 + K31$, $K29 + K32$)
二重押しが確定した場合, キー・データHは9Hを設定します。
 $K29 + K30$...参照アドレス = 19DH
 $K29 + K31$...参照アドレス = 19EH
 $K29 + K32$...参照アドレス = 19FH

6.2.2 ディテール・フロー・チャート



☆☆☆☆☆☆☆☆ キー入力処理 ☆☆☆☆☆☆☆☆

```

TIME9M EQU 1FFH ;9.00ms(9.002ms)
CUSTM1 EQU 050H ;Custom Code = 0AH
CUSTM2 EQU 0AFH ;Custom Code' = F5H

```

```

;##### P U B L I C #####
PUBLIC KEY48
;

```

```

;##### E X T E R N #####
EXTRN MAIN ;MAIN Routine
;

```

```

;##### S T A R T #####
KEY48:
MOV RO,#00CH

```

① チャタリング・カウンタ (R₀₀) を設定 (=0CH) します (0FHで3回カウント終了)。

```

;*****
; ON Chattering
;*****

```

```

ON_CHAT:
STTS #0101B
OUT PO,#0FFH
OUT P4,#033H
HALT #005H
MOV T,#TIME9M
MOV R9,#000H
MOV R6,#008H

```

- ② ステータス・フラグをクリアします。
- ③ キー・スキャン出力 (K₁₀) をすべてハイ・レベルに設定します。
- ④ S₁, S₀を入力モードに設定します。
- ⑤ HALTモード: タイマ・カウンタが00Hになると解除されます。
- ⑥ タイマ・カウンタを設定します。9ms=1FFH
- ⑦ 比較キー・データ (R₉) のイニシャライズ (00H) を行います。
- ⑧⑨ キー・スキャン・カウンタ (R₀₀) を8H, キーありフラグ (R₁₀) を0Hに設定します。

```

STTS #1110B
JNF MAIN
MOV R8,#001H

```

- ⑩ キー入力 (K₁, S₀, S₁) があるか判断します。キー入力なしの場合, MAINへ
- ⑪ キー・スキャン・データ (R₈) のイニシャライズ (01H) を行います。

```

;*****
; Key Scan
;*****

```

```

KEYS_OUT:
MOV A,R8
OUT P00,A
MOV A,R8
OUT P10,A

```

⑫ キー・スキャン・データ (P₀) を出力します。

```

;+++++
; ++ キー入力まで100μS Wait ++
;+++++

```

```

MOV R2,#0E1H
MOV RB,#0C3H
NOP
NOP
NOP
NOP

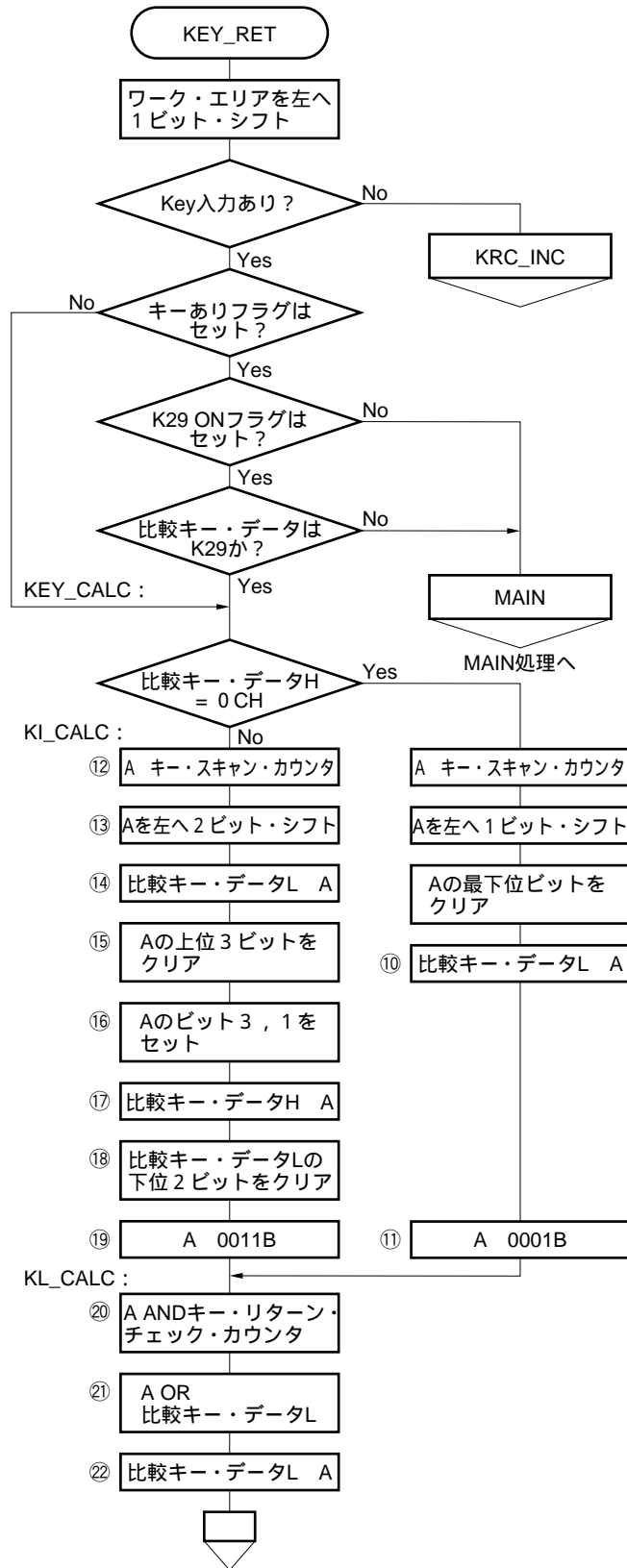
```

- [備考] R2をイミディエト・データ設定用に使用 (R₁₂: 0EH, R₀₂: 1H) します。
- [備考] RBをイミディエト・データ設定用に使用 (R_{1B}: 0CH, R_{0B}: 3H) します。
- ⑬ キー入力チェックを行うまで100μs待ちます。

```

;*****
;           Key Return Check
;*****
      STTS  #1011B           ⑭ キー入力 (Ki, S0, S1) があるかどうかを判断します。
      JNF   NO_KEYS         キー入力なしの場合, NO_KEYSへ
      STTS  #0011B           ⑮ Ki入力かS0, S1入力かを判断します。
      JNF   SOS1CALC        S0, S1の場合, SOS1CALCへ
;*****
;           KI Data Calculate
;*****
      IN    A, P01           ⑯ Ki入力があり, S0またはS1入力もあるかどうか判断します。
      RL    A                S0またはS1入力がある場合, 多重押しエラーと判断し, MAINへ
      JC    MAIN             ⑰ キー・リターン・チェック・カウンタ (R0A) イニシャライズを行います。
      RL    A                0CH=4回 (このとき R1B=0CH)
      JC    MAIN             ⑱ アキュムレータにKiの端子状態 (P11) を転送します。
      MOV   A, R1B
      MOV   R0A, A
      IN    A, P11
      JMP   KEY_RETO
;*****
;           S0,S1 Data Calculate
;*****
SOS1CALC:
      MOV   R9, #0C0H        ⑲ 比較キー・データH (R19) にK33~K48 (S0, S1入力) 判断用の値 (0CH) を設定
                              します。
      MOV   A, R12
      MOV   R0A, A           ⑳ キー・リターン・チェック・カウンタ (R0A) のイニシャライズを行います。
                              0EH=2回用 (このとき R12=0EH)
      IN    A, P01           ㉑ アキュムレータにS0, S1の端子状態 (P01) を転送します。
KEY_RETO:
      MOV   R10, A           [備考] ワーク・エリアにアキュムレータの値 (P00またはP01) を格納します。
      STTS  #0101B         ㉒ ステータス・フラグをクリアします。

```



KEY_RET:

```

MOV  A, R10 }
RLZ  A      }----- ① ワーク1 (R10) を左へ1ビット・シフトします。
MOV  R10, A }
JNC  KRC_INC }----- ② キー入力がある (CY=1) かを判断します。
; ** Key入力あり **                               キー入力がない場合, KRC_INCへ
MOV  A, R16 }
SCAF A, R16 }----- ③ キーが押されている (二重押し) かを判断します。
JNC  KEY_CALC }     キーが押されていない (単押し) 場合, KEY_CALCへ
MOV  A, R13 }
SCAF A, R13 }----- ④ K29 ONフラグがセットされているかを判断します。
JNC  MAIN }         クリアされている場合, MAINへ
MOV  A, #0100B }
XRL  A, R19 }----- ⑤ 比較キー・データ (R9) はK29のキー・データと一致するかを判断します。
SCAF A, R19 }     一致していない場合, MAINへ
JNC  MAIN }
MOV  A, R09 }
XRL  A, R0B }
SCAF A, R0B }
JNC  MAIN }

```

```

;*****
:      Key Data Calculate
;*****

```

KEY_CALC:

```

MOV  A, R0B }----- ⑥ 押されたキーがK33~K48 (S0, S1入力) かどうかを判断します。
XRL  A, R19 }     K1~K32のキーが押された場合,
SCAF A, R19 }     KI_CALCへ
JNC  KI_CALC }

```

; ** S0, S1データ算出 **

```

MOV  A, R06 }----- ⑦ キー・スキャン・カウンタ (R06) をアキュムレータに転送します。
RL   A }----- ⑧ アキュムレータを左へ1ビット・シフトします。
ANL  A, R12 }----- ⑨ アキュムレータの最下位ビットをクリアします。
MOV  R09, A }----- ⑩ アキュムレータの値を比較キー・データL (R09) に格納します。
MOV  A, R02 }----- ⑪ アキュムレータに0001Bを設定します
JMP  KL_CALC }----- KL_CALCへ

```

アキュムレータの値

KS3	KS2	KS1	KS0
-----	-----	-----	-----

KS2	KS1	KS0	KS3
-----	-----	-----	-----

KS2	KS1	KS0	0
-----	-----	-----	---

			R09
--	--	--	-----

0	0	0	1
---	---	---	---

; ** KIデータ算出 **

KI_CALC:

```

MOV  A, R06 }----- ⑫ キー・スキャン・カウンタ (R06) をアキュムレータに転送します。
RL   A }----- ⑬ アキュムレータを左へ2ビット・シフトします。
RL   A }
MOV  R09, A }----- ⑭ アキュムレータの値を比較キー・データL (R09) に格納します。
ANL  A, R02 }----- ⑮ アキュムレータの上位3ビットをクリアします。
ORL  A, #1010B }----- ⑯ アキュムレータのビット3, ビット1をセット
MOV  R19, A }----- ⑰ アキュムレータの値を比較キー・データH (R19) に格納します。
MOV  A, R09 }----- ⑱ 比較キー・データL (R09) の下位2ビットをク
ANL  A, R1B }     リアします。 (このときR1B=0CH)
MOV  R09, A }-----
MOV  A, R0B }----- ⑲ アキュムレータに0011Bを設定します。

```

KS3	KS2	KS1	KS0
-----	-----	-----	-----

KS1	KS0	KS3	KS2
-----	-----	-----	-----

			R09
--	--	--	-----

0	0	0	KS2
---	---	---	-----

1	0	1	KS2
---	---	---	-----

			R19
--	--	--	-----

KS1	KS0	0	0
-----	-----	---	---

			R09
--	--	--	-----

0	0	1	1
---	---	---	---

アキュムレータの値

KL_CALC:

ANL A, R0A

- ⑩ K_i入力の場合、アキュムレータに キー・リターン・カウンタ (R0A) の下位2ビットを、S₀, S_i入力の場合、下位1ビットを格納します。

K_i入力の場合

0	0	KR1	KR0
---	---	-----	-----

S_i, S₀入力の場合

0	0	0	KR0
---	---	---	-----

ORL A, R09

- ⑪ アキュムレータと比較キー・データL (R09) を ORします。

K_i入力の場合

KS1	KS0	KR1	KR0
-----	-----	-----	-----

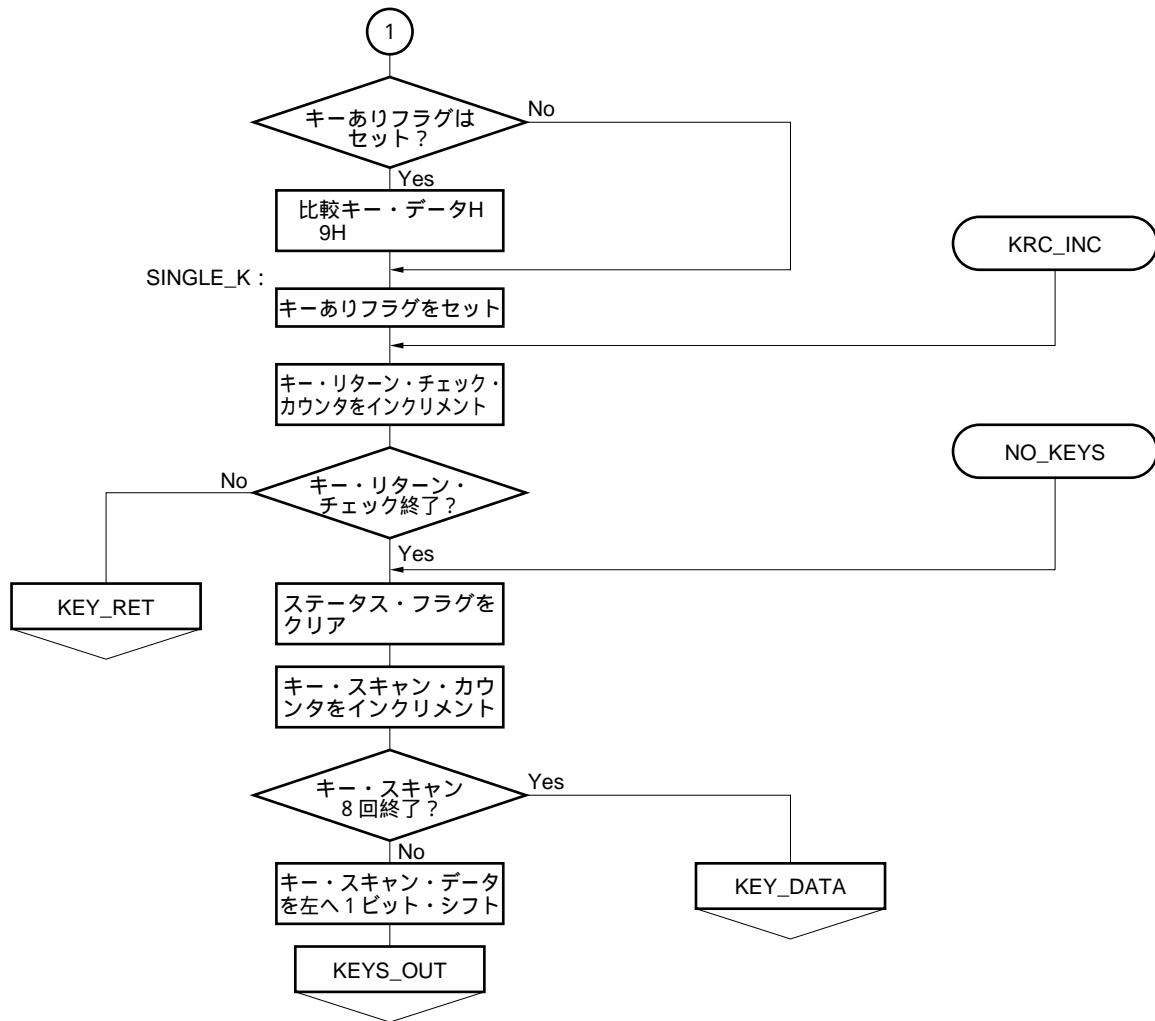
S_i, S₀入力の場合

KS2	KS1	KS0	KR0
-----	-----	-----	-----

MOV R09, A

- ⑫ アキュムレータの値を比較キー・データL (R09) に格納します。

R09



```

; ** 二重押し Check **
MOV   A, R16
SCAF  SINGLE_K
JNC   SINGLE_K
MOV   A, #09H
MOV   R19, A
SINGLE_K:
MOV   A, #0FH
MOV   R16, A
KRC_INC:
MOV   A, R0A
INC   A
MOV   R0A, A
JNC   KEY_RET
NO_KEYS:
STTS  #0101B
MOV   A, R06
INC   A
MOV   R06, A
JC    KEY_DATA
MOV   A, R18
RL    A
MOV   R18, A
MOV   A, R08
RL    A
MOV   R08, A
JNC   KEYS_OUT
MOV   R8, #010H
JMP   KEYS_OUT

```

① キーありフラグ (R16) がセットされているかを判断します。
クリアの場合、SINGLE_Kへ

② 比較キー・データH (R19) に9Hを設定します。

③ キーありフラグ (R16) をセットします (このときアキュムレータ=0FH)。

④ キー・リターン・チェック・カウンタ (R0A) をインクリメントします。

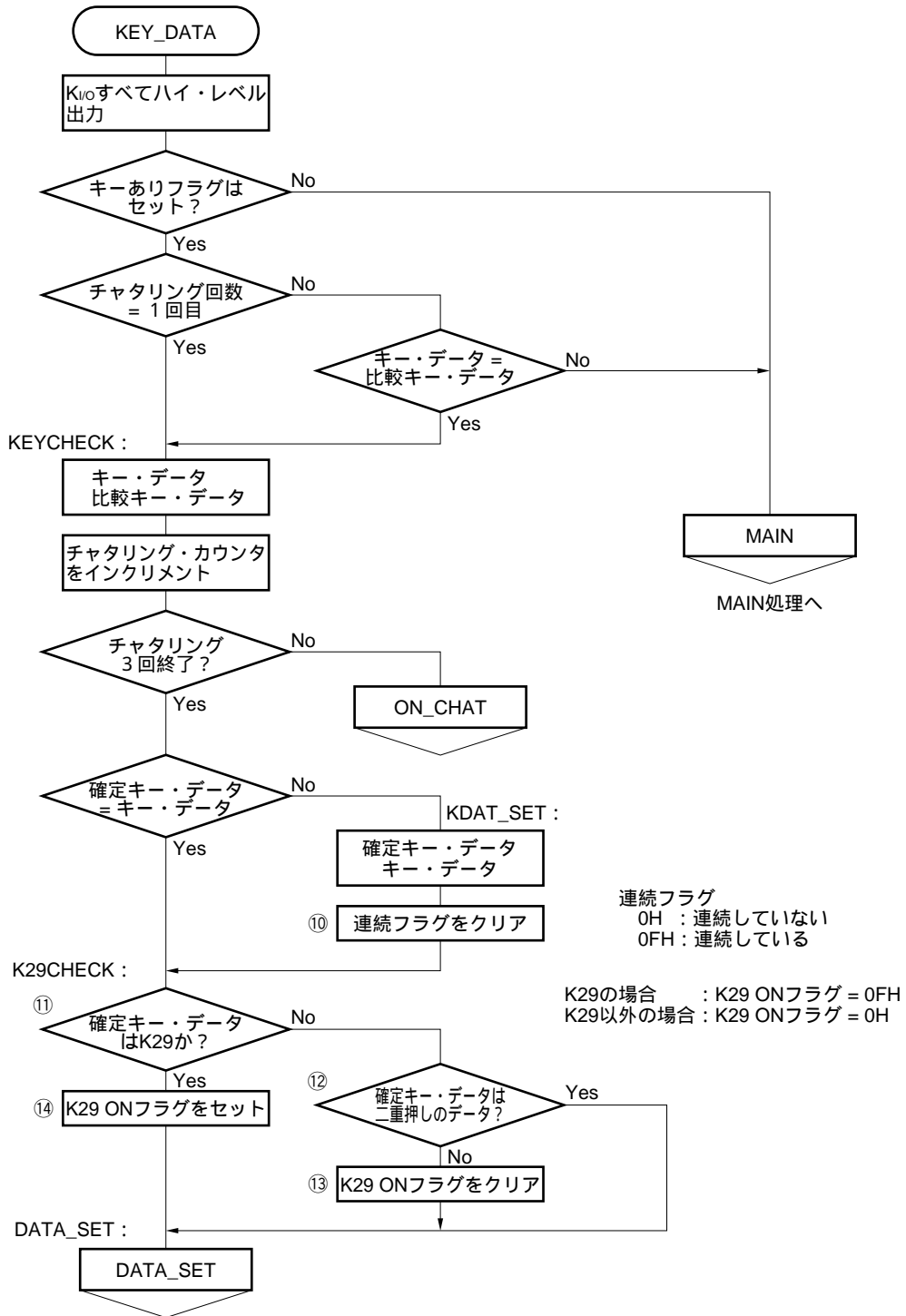
⑤ キー・リターン・チェックが4回終了したかを判断します。
まだ終了していない場合、KEY_RETへ

⑥ ステータス・フラグをクリアします。

⑦ キー・スキャン・カウンタ (R06) をインクリメントします。

⑧ キー・スキャンが8回終了したかを判断します。
終了の場合、KEY_DATAへ

⑨ キー・スキャン・データ (R8) を左へ1ビット・シフトします。



```

;*****
; Transmit Key Data = ON Chattering Key Data ??
;*****
KEY_DATA:
    OUT    PO,#OFFH          ① キー・スキャン出力 (K170) をすべてハイ・レベルに設定します。
    MOV    R8,#0FOH        [備考] R8をイミディエト・データ設定用に使用 (R18:0FH, R08:0H) します。
    MOV    A,R16
    SCAF   }
    JNC   MAIN             }----- ② キーありフラグ (R18) がセットされているかを判断します。
                             } クリアの場合、MAINへ
    MOV    A,R00
    XRL   A,R0B             }----- ③ チャタリング処理が1回目 (0CH) か判断します。
    SCAF   }                } (このときR08=#0011B)
    JC    KEYCHECK         } 1回目の場合、KEYCHECKへ
    MOV    A,R14
    XRL   A,R18             }----- [備考] このときR18=0FH
    XRL   A,R19
    SCAF   }
    JNC   MAIN             }----- ④ キー・データ (R4) と比較キー・データ (R9) が一致しているかを判断しま
    XRL   A,R04             } す。
    XRL   A,R09             } (このときアキュムレータ=0FH)
    SCAF   }                } 不一致の場合、MAINへ
    JNC   MAIN
KEYCHECK:
    MOV    A,R19
    MOV    R14,A           }----- ⑤ 比較キー・データ (R9) をキー・データ (R4) に格納します。
    MOV    A,R09
    MOV    R04,A
    MOV    A,R00
    INC    A               }----- ⑥ チャタリング・カウンタ (R00) をインクリメントします。
    MOV    R00,A
    SCAF   }
    JNC   ON_CHAT         }----- ⑦ チャタリング処理が3回終了したかを判断します。
; ** Key High Check **   } まだ終了していない場合、ON_CHATへ
    XRL   A,R11
    XRL   A,R14           }----- [備考] このときアキュムレータ=0FH
    SCAF   }
    JNC   KDAT_SET
; ** Key Low Check **   }----- ⑧ 確定キー・データ (R1) とキー・データ (R4) は一致しているかを判断しま
    XRL   A,R01           } す。
    XRL   A,R04           } (このときアキュムレータ=0FH)
    SCAF   }                } 不一致の場合、KDAT_SETへ
    JC    K29CHECK        } 一致の場合、K29CHECKへ
KDAT_SET:
    MOV    A,R14
    MOV    R11,A          }----- ⑨ キー・データ (R4) を確定キー・データ (R1) に格納します。
    MOV    A,R04
    MOV    R01,A
    MOV    A,R08
    MOV    R03,A          }----- ⑩ 連続フラグ (R03) をクリアします (このときR08=0H)。
K29CHECK:
    MOV    A,#0100B
    XRL   A,R11
    SCAF   }
    JNC   DATA_K_CHK     }----- ⑪ K29が押されたかを判断します。
    MOV    A,R01           } (このときR08=3H)
    XRL   A,R0B           } K29以外のキーが押された場合、K29_FLGへ
    SCAF   }
    JNC   K29_FLG

```

```

DBL_K_CHK:
  MOV   A, #06H
  XRL   A, R11
  SCAF
  JC    DATA_SET

```

⑫ 確定キー・データは二重押しのデータかを判断します。
二重押しのデータの場合、DATA_SETへ

```

  MOV   A, R08

```

⑬ アキュムレータに0Hをセットします（このときR₀₈=0H）。

```

K29_FLG:
  MOV   R13, A

```

⑬⑭ K29 ONフラグをクリアします（このときアキュムレータ=0H）。
K29 ONフラグをセットします（このときアキュムレータ=0FH）。



```

;*****
;          Transmit Data Set
;*****
DATA_SET:
  ;** Custom Code **
  MOV     R4,#CUSTM1           ① カスタム・コード (R4) にテーブル・データを格納します。

  ;** Custom Code' **
  MOV     R5,#CUSTM2           ② カスタム・コード' (R5) にテーブル・データを格納します。

  ;** Data Code **
  MOV     A,R11                }
  MOV     R10,A                }----- ③ 確定キー・データ (R1) をデータ・ポインタ (R0) に設定します。
  MOV     A,R01                }
  MOV     R00,A                }
  MOV     R6,@R0               ④ データ・コード (R6) にテーブル・データを格納します。
END

```

6.3 送信処理

送信処理は、データ・メモリに格納されたカスタム・コード、カスタム・コード'、データ・コード、データ・コードを第2章 送信波形で示したNEC-Rフォーマットに従い送信します。

送信方法は、送信時間をデータ・テーブルで持ち、テーブル参照によりタイマ・カウンタに設定します。その後、HALTモード（解除条件：TIMER）に入ることにより送信が可能になります。

時間管理を正確および簡単に行うために、図6-10のような方法を行います。HALT解除直後に、次の送信時間データを上記と同様の方法でタイマ・カウンタに設定することにより、HALT-HALT間が設定時間で動作することになります。

キャリア出力あり（以下‘H’とする）を設定するときには、タイマ・カウンタの最上位ビット（出力制御ビット）をセットします。また、キャリア出力なし（以下‘L’とする）を設定するときには、タイマ・カウンタの最上位ビット（出力制御ビット）をクリアします。

タイマ・カウンタの構成を図6-9に示します。

図6-9 タイマ・カウンタの構成

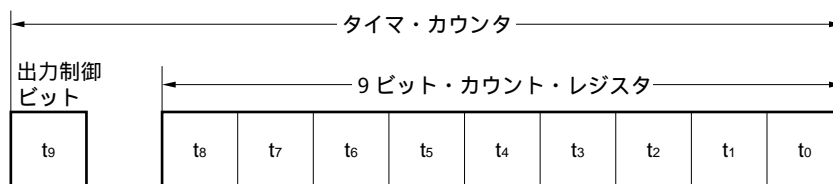
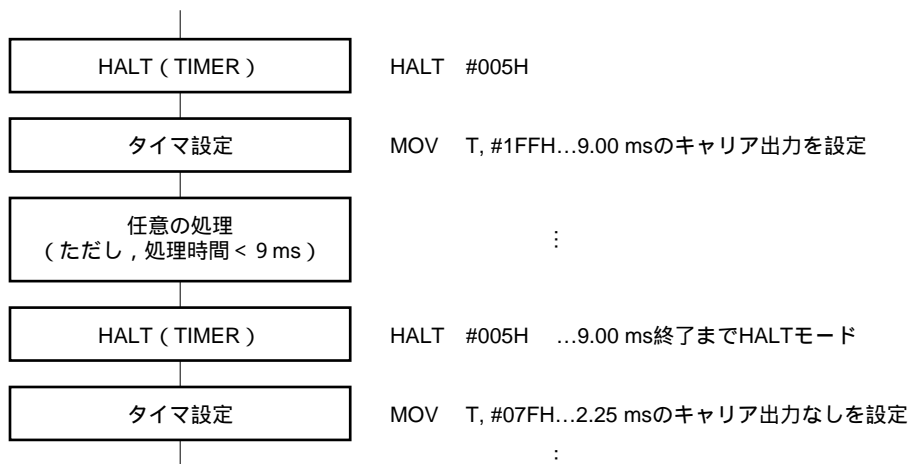


図6-10 タイマ・カウンタの時間管理



タイマの動作時間は「(設定値 + 1) × 8 / fx」で求めることができます。

455 kHzで動作している場合、「MOV T,#1FFH」は「(1FFH + 1) × 8 / (455 × 10³) ÷ 9.00 ms」を設定したことになります。

6.3.1 処理説明

送信処理は次に示す7つの処理により構成されています。

- ・初期設定
- ・リーダー・コードの送信
- ・コード送信(カスタム・コード, カスタム・コード', データ・コード, データ・コード)
- ・ストップ・ビットの送信
- ・フレーム・スペースの送信
- ・2フレーム以降の送信
- ・オフ・チャタリング除去処理

次に、各処理について説明します。

(1) 初期設定

送信処理の初期設定内容を次の , に示します。

キー・オフ・チェック・カウンタ(R07)を10回(=05H)に設定します。

データ1送信カウンタを0DFH(24ビット用)に設定します。

カウンタの内訳を表6-5に示します。

このデータ1送信カウンタは、本来フレーム・スペースの送信時間に影響を与えるカスタム・コード, カスタム・コード'内の'データ1'の個数をカウントするためのカウンタです。

データ・コード, データ・コード内の'データ1'の個数は固定(8個)のためフレーム・スペースの送信時間に影響を与えません。

表6-5 データ1送信カウンタ(R9)の内訳

カウンタ値	データ1送信カウンタの内訳
D0H E6H	'データ1'の個数: 0~7個 データ・コード, データ・コード内の'データ1'の個数は固定(8個)のため, この状態は存在しない
E7H	'データ1'の個数: 8個 カスタム・コード, カスタム・コード'の'データ1'の個数: 0個
E8H EFH	'データ1'の個数: 9~16個 カスタム・コード, カスタム・コード'の'データ1'の個数: 1~8個
F0H F7H	'データ1'の個数: 17~24個 カスタム・コード, カスタム・コード'の'データ1'の個数: 9~16個

(2) リーダ・コードの送信

リーダー・コードの送信方法を次の ~ に示します。

リーダー・コード用のテーブル参照用アドレス (1D8H) を次のように設定します。

- ・データ・ポインタH (R10) に上位アドレス (0DH)
- ・データ0 (R08) にビット・データの 'データ0' の下位アドレス (8H)

リーダー・コード用の出力データ0Hをワーク2 (R1A) に設定します。

リーダー・コードを1ビット送信としてサブルーチン (BITOUT0F) をコールします。

ワーク2 (R1A) を左シフトさせてビット判断を行います。

判断されたビットのテーブル参照用アドレスLを、データ・ポインタLに設定することにより、送信時間をタイマ・カウンタに設定することができます。

リーダー・コード (H : 9.00 ms , L : 4.50 ms) を送信します。

(3) コード送信 (カスタム・コード, カスタム・コード', データ・コード, データ・コード)

コード送信方法を次の ~ に示します。

コード送信用ビット・データのテーブル参照用アドレス (1DAH, 1DCH) を次のように設定します。

- ・データ・ポインタH (R10) に上位アドレス (0DH)
- ・データ0 (R08) にビット・データの 'データ0' の下位アドレス (0AH)
- ・データ1 (R18) にビット・データの 'データ1' の下位アドレス (0CH)

次に示すデータを各コード送信時に、ワーク2 (R1A) に設定します。

- ・カスタム・コードH (R14)
- ・カスタム・コードL (R04)
- ・カスタム・コードH' (R15)
- ・カスタム・コードL' (R05)
- ・データ・コードH (R16)
- ・データ・コードL (R06)
- ・データ・コードH (R16を全反転する)
- ・データ・コードL (R06を全反転する)

各コード送信を4ビット送信としてサブルーチンをコールします。

ワーク2 (R1A) を左シフトさせてビット判断を行います。このときに 'データ1' が判断された場合、フレーム・スペースの送信のため、'データ1' の個数をカウントします。

判断されたビットのテーブル参照用アドレスLを、データ・ポインタLに設定することにより、送信時間をタイマ・カウンタに設定することができます。

各コードを送信します。

- ・ 'データ0' ... H : 0.56 ms , L : 0.56 ms
- ・ 'データ1' ... H : 0.56 ms , L : 1.69 ms

このとき、1ビットごとのロウ・レベル送信中にキー・オフのチェックを行います。

キー・オフのチェックについては、(7) オフ・チャタリング除去処理を参照してください。

4ビット分の送信が終了するまで ~ を繰り返します。

カスタム・コードHからデータ・コードLの送信が終了するまで ~ を繰り返します。

(4) ストップ・ビットの送信

ストップ・ビットの送信方法を次の ~ に示します。

ストップ・ビット用のテーブル参照用アドレス (1DEH) を次のように設定します。

- ・データ・ポインタH (R10) に上位アドレス (0DH)
- ・データ0 (R08) にビット・データの 'データ0' の下位アドレス (0EH)

ストップ・ビット用の出力データ0Hをワーク2 (R1A) に設定します。

ストップ・ビットを1ビット送信としてサブルーチンをコールします。

ワーク2 (R1A) を左シフトさせてビット判断を行います。

判断されたビットのテーブル参照用アドレスLを、データ・ポインタLに設定することにより、送信時間をタイマ・カウンタに設定することができます。

ストップ・ビット (H : 0.56 ms, L : 3.00 ms) を送信します。

(5) フレーム・スペースの送信

コード送信時間は、ビット・データの 'データ1' と 'データ0' で送信時間に差があるため、59.06 ms ~ 77.06 ms の範囲で変化します。そのためコード送信を行う場合には、フレーム・スペースの送信時間が、'データ1' の個数によって変化します。

ただし、データ・コード、データ・コード内の 'データ1' の個数は固定 (8個) のためフレーム・スペースの送信時間には影響しません。フレーム・スペースの送信時間に影響を与えるのは、カスタム・コード、カスタム・コード内の 'データ1' の個数です。

送信時間の変化に対応するため、送信時に 'データ1' の個数をカウントします。このカウント値によりフレーム・スペースの送信時間を調節します。'データ1' のカウント処理は (3) コード送信を参照してください。

'データ1' の個数に対するフレーム・スペースの時間を表6-6に示します。

また、タイマ・カウンタは設定可能な最大時間が9.00 msのため、図6-11 ~ の3パターンのように、フレーム・スペースを分割して送信を行います。

表6-6と図6-11との差は、コード送信を行ったときの誤差 ('データ0' のロウ・レベル = 0.565 msを0.56 msに設定) をフレーム・スペースで吸収するためです。

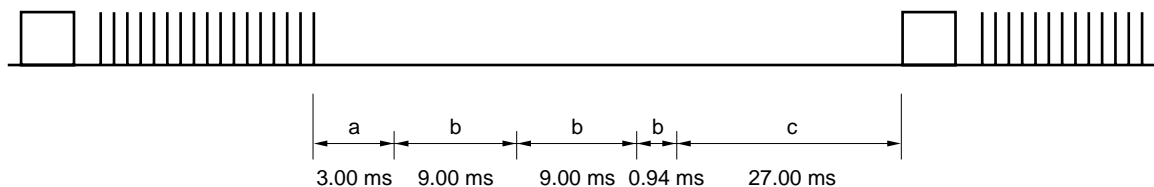
表6-6 'データ1'の個数に対するフレーム・スペースの時間

データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)
8	48.940	13	43.315	18	37.690	23	32.065
9	47.815	14	42.190	19	36.565	24	30.940
10	46.690	15	41.065	20	35.440		
11	45.565	16	39.940	21	34.315		
12	44.440	17	38.815	22	33.190		

備考 データ・コード, データ・コード内の'データ1'の個数は固定(8個)のため, 'データ1'が0~7個の場合はありません。

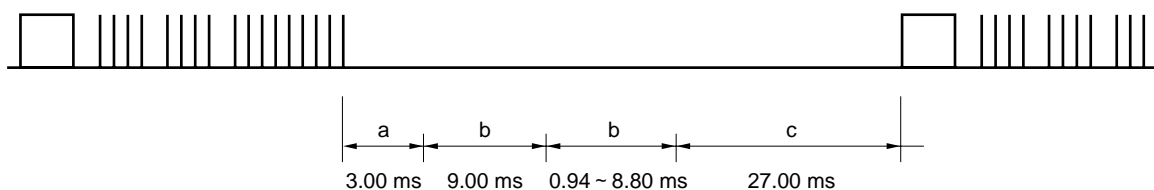
図6-11 フレーム・スペース送信時間の分割方法(1/2)

'データ1'が8個(データ1送信カウンタ値:0E7H)の場合



- a: ストップ・ビット送信後, フレーム・スペースを3.00 ms送信します。
- b: 'データ1'の個数に対する時間(18.94 ms), フレーム・スペースを送信します。
- c: キーを押し続けた場合には, オン・チャタリング除去処理(9.00 ms x 3回)を行います。
それ以外の場合には, フレーム・スペースを27.00 ms送信します。

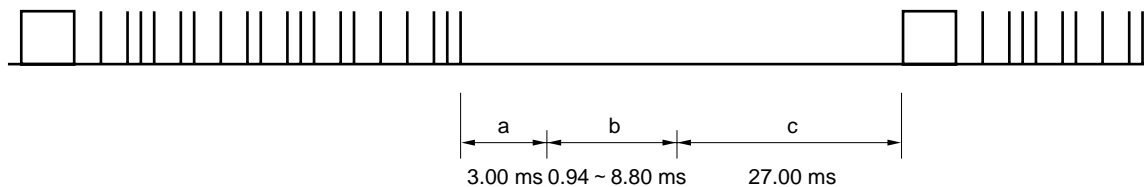
'データ1'が9~16個(データ1送信カウンタ値:0E8H~0EFH)の場合



- a: ストップ・ビット送信後, フレーム・スペースを3.00 ms送信します。
- b: 'データ1'の個数に対する時間(9.94~17.80 ms), フレーム・スペースを送信します。
- c: キーを押し続けた場合には, オン・チャタリング除去処理(9.00 ms x 3回)を行います。
それ以外の場合には, フレーム・スペースを27.00 ms送信します。

図6-11 フレーム・スペース送信時間の分割方法(2/2)

‘データ1’が17~24個(データ1送信カウンタ値:0F0H~0F7H)の場合



- a : ストップ・ビット送信後, フレーム・スペースを3.00 ms送信します。
- b : ‘データ1’の個数に対する時間(0.94~8.80 ms), フレーム・スペースを送信します。
- c : キーを押し続けた場合には, オン・チャタリング除去処理(9.00 ms × 3回)を行います。
それ以外の場合には, フレーム・スペースを27.00 ms送信します。

(6) 2フレーム以降の送信

キーが離されるかキーが変更されるまで, 1フレーム目と同様のコードの送信を, リーダ・コードから繰り返します。送信方法は, (2)~(5)を参照してください。

キー・オフのチェックについては, (7)オフ・チャタリング除去処理を参照してください。

(7) オフ・チャタリング除去処理

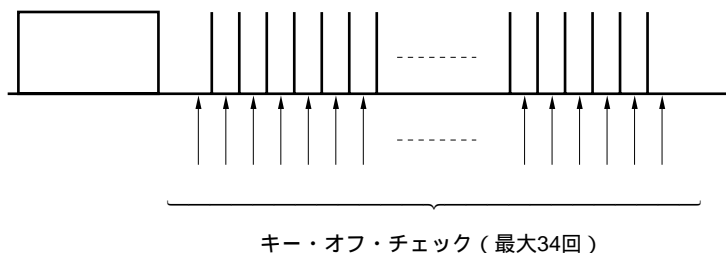
ビット・データのロウ・レベル出力中にキー・オフのチェックを行います。

リーダー・コード, ストップ・ビットを含め34回のロウ・レベル出力中に, 10回連続でキー入力なしと判断した場合のみ, キー・オフが確定します。10回連続チェック中に1回でもキー入力ありと判断した場合には, 10回連続用のカウンタはクリアされ, 最初からカウントを開始します。

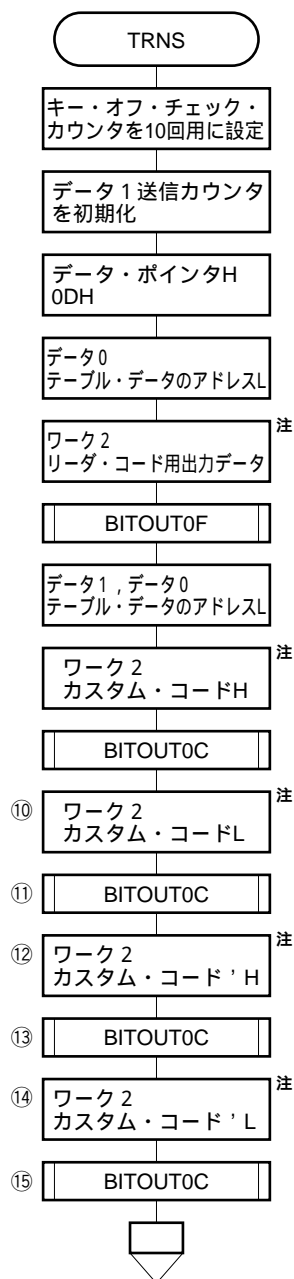
34回のチェックでキー・オフが確定しなかった場合, キーが押し続けられていると判断します。

1フレーム送信中にキー・オフが確定した場合でも, 最低2フレームの送信を行ってから初期設定処理を行います。

図6-12 ビット・データ送信時のキー・オフ・チェック



6.3.2 ディテール・フロー・チャート



注 メイン・ルーチンでは、リーダ・コード出力データをアキュムレータ（A）に設定し、サブルーチン（BITOUT0C）でワーク2（R1A）にそのデータを設定します。

☆☆☆☆☆☆☆☆ 送信処理 ☆☆☆☆☆☆☆☆

```

TIME9M EQU 1FFH ;9.00ms (9.002ms)

;##### P U B L I C #####
;

;##### E X T E R N #####
EXTRN MAIN ;MAIN Routine
EXTRN KEY48 ;Key Check Routine
;

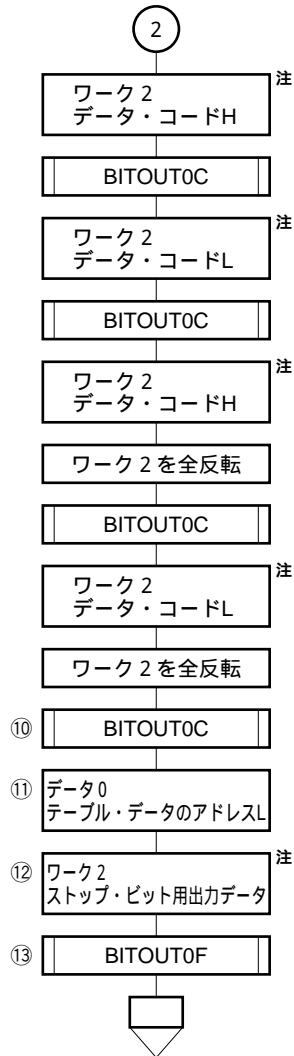
;##### S T A R T #####
TRNS:
MOV R7,#0F5H ① キー・オフ・チェック・カウンタ (R07) を10回 (=5H) に設定します。
                R17をイミューディエト設定用に使用 (R17=0FH) します。
MOV R9,#0DFH ② データ1送信カウンタ (R9) をイニシャライズ (DFH) します。

;*****
; Leader Code [備考] リード・コードの送信 (H:9.00ms, L:4.50ms)
;*****
MOV R0,#0D0H ③ データ・ポインタH (R10) に0DH (ビット・データ送信時間用テーブル・
                データの上位アドレス) を設定します。
MOV R8,#008H ④ データ1 (R18), データ0 (R08) にビット・データ送信時間用テーブル・
                データの低位アドレスを設定します (R18←0H, R08←8H)。
MOV A,R18 ⑤ アキュムレータにリード・コード用の出力データ (0H) を設定します
                (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL BITOUT0F ⑥ ビット出力サブルーチンをコールします (1ビット送信用:カウンタ値=0FH)。

;*****
; Custom Code [備考] カスタム・コードの送信 (H:0.56ms, L:0.56msまたは1.69ms)
;*****
MOV R8,#0CAH ⑦ データ1 (R18), データ0 (R08) にカスタム・コード用のテーブル・データの低位
                アドレスを設定します (R18←CH, R08←AH)。
MOV A,R14 ⑧ アキュムレータにカスタム・コード用の出力データH (R14) を設定します
                (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL BITOUT0C ⑨ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
MOV A,R04 ⑩ アキュムレータにカスタム・コード用の出力データL (R04) を設定します
                (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL BITOUT0C ⑪ ビット出力サブルーチンをコールします
                (4ビット送信用:カウンタ値=0CH)。

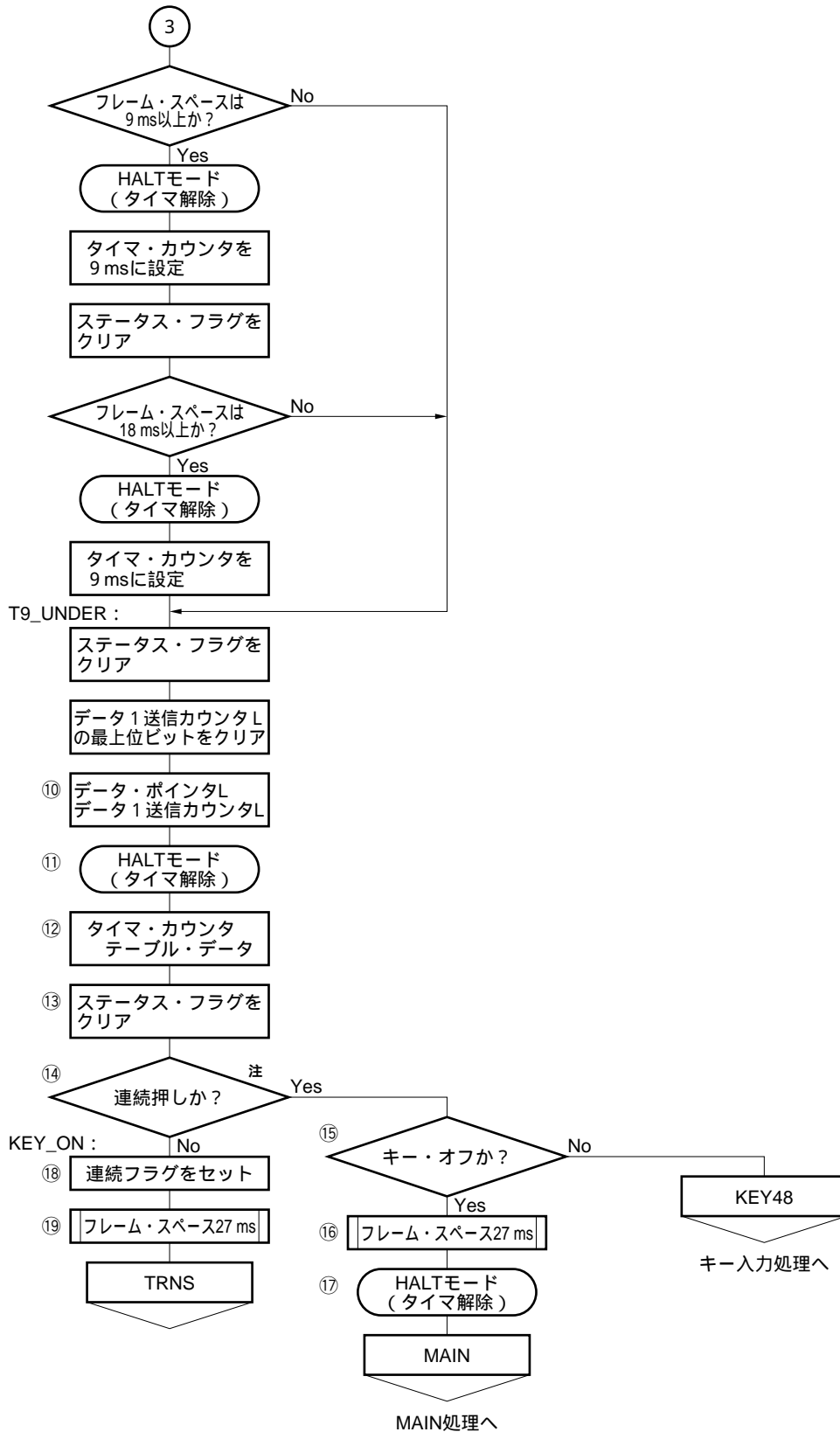
;*****
; Custom Code' [備考] カスタム・コード' の送信 (H:0.56ms, L:0.56msまたは1.69ms)
;*****
MOV A,R15 ⑫ アキュムレータにカスタム・コード' 用の出力データH (R15) を設定します
                (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL BITOUT0C ⑬ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
MOV A,R05 ⑭ アキュムレータにカスタム・コード' 用の出力データL (R05) を設定します
                (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL BITOUT0C ⑮ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。

```



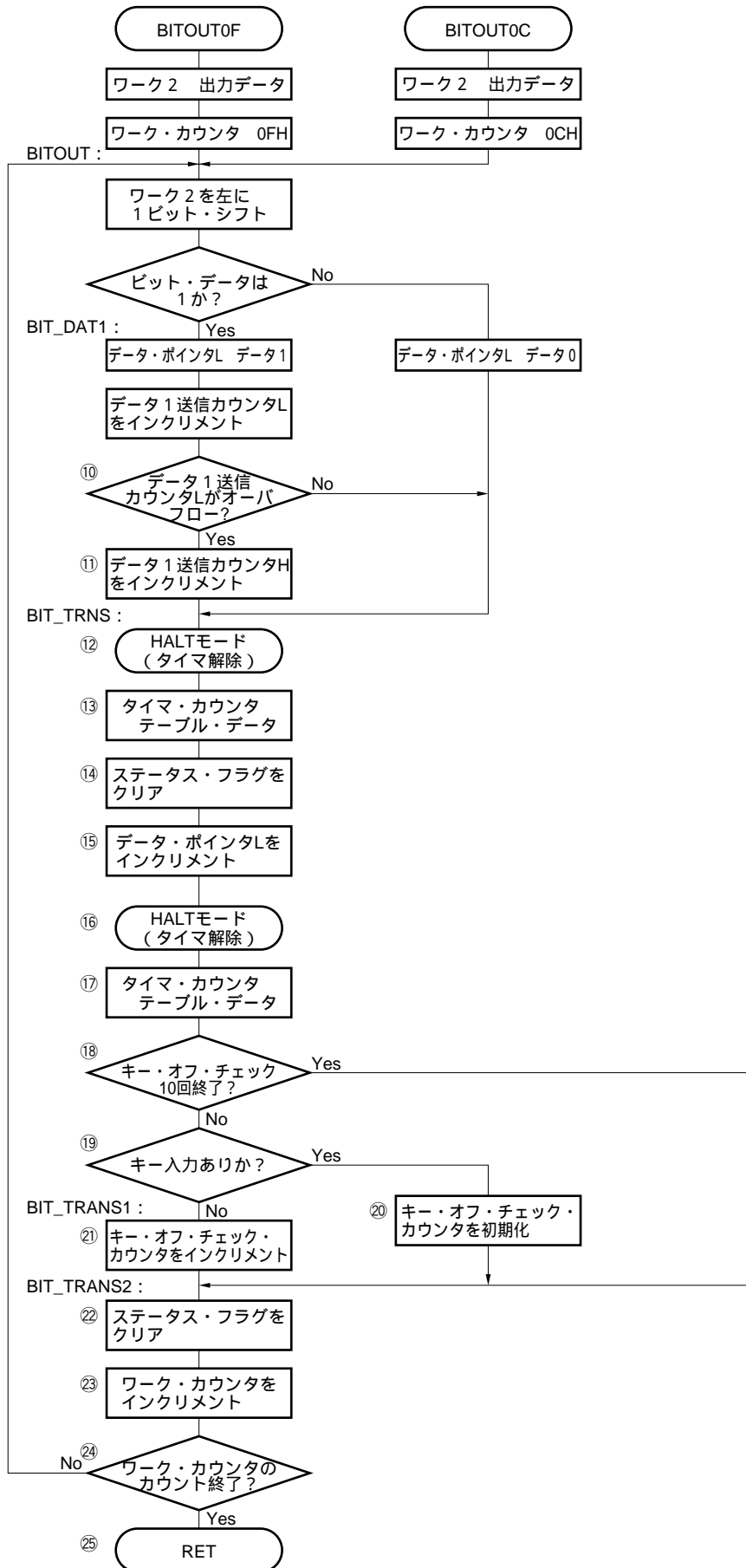
注 メイン・ルーチンでは、リーダー・コード用出力データをアキュムレータ（A）に設定し、サブルーチン（BITOUT0C）でワーク2（R1A）にそのデータを設定します。

;*****		
;	Data Code	[備考] データ・コードの送信 (H: 0.56ms, L: 0.56msまたは1.69ms)
;*****		
MOV	A, R16	① アキュムレータにデータ・コード用の出力データH (R16) を設定します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	② ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。
MOV	A, R06	③ アキュムレータにデータ・コード用の出力データL (R06) を設定します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	④ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。
;*****		
;	Data Code'	[備考] $\overline{\text{データ・コード}}$ の送信 (H: 0.56ms, L: 0.56msまたは1.69ms)
;*****		
MOV	A, R16	⑤ アキュムレータにデータ・コード用の出力データH (R16) を設定します。
XRL	A, #0FH	⑥ アキュムレータの値を反転させ、 $\overline{\text{データ・コードH}}$ の出力データを作成します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	⑦ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。
MOV	A, R06	⑧ アキュムレータにデータ・コード用の出力データL (R06) を設定します。
XRL	A, #0FH	⑨ アキュムレータの値を反転させ、 $\overline{\text{データ・コードL}}$ の出力データを作成します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	⑩ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。
;*****		
;	Stop Bit	[備考] ストップ・ビットの送信 (H: 0.56ms, L: 3ms)
;*****		
MOV	R8, #00EH	⑪ データ0 (R08) にストップ・ビットのキャリアありのテーブル・データ下位 アドレスを設定します。
MOV	A, R18	⑫ アキュムレータにストップ・ビット用の出力データ (0H) を設定します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0F	⑬ ビット出力サブルーチンをコールします (1ビット送信用: カウンタ値=0FH)。



注 連続押しであれば，連続フラグ = 0FH

;*****		
;	Frame Space	[備考] フレーム・スペースの送信 (L:30.94ms~48.94ms)
;*****		
MOV	A, R19	① カスタム・コード (16ビット) のビット・データ '1' の数が8個以下か判断 します。 9個以上の場合, T9_UNDERへ
SCAF		
JC	T9_UNDER	
HALT	#005H	② HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV	T, #TIME9M	③ 8個以下の場合, タイマ・カウンタに9ms (=1FFH) を設定します。
STTS	#0101B	④ ステータス・フラグをクリアします。
MOV	A, R09	⑤ カスタム・コード (16ビット) のビット・データ '1' の数が, 0個かを判断 します。 1~8個の場合, T9_UNDERへ
RL	A	
JC	T9_UNDER	
HALT	#005H	⑥ HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV	T, #TIME9M	⑦ 0個の場合, 計18ms送信するためにタイマ・カウンタに再度9ms (=1FFH) を設定 します。
T9_UNDER:		
STTS	#0101B	⑧ ステータス・フラグをクリアします。
MOV	A, R09	⑨ データ1送信カウンタL (R09) の最上位ビットをクリアします。
ANL	A, #0111B	
MOV	R00, A	⑩ データ1送信カウンタL (R09) の値をデータ・ポインタL (R00) に設定しま す。
HALT	#005H	⑪ HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV	T, @R0	⑫ テーブル参照により, タイマ・カウンタにビット・データ '1' の数に応じて 0.940ms~8.815msを設定します。
STTS	#0101B	⑬ ステータス・フラグをクリアします。
MOV	A, R03	⑭ 1フレーム目の送信 (連続押しではないか) を判断します。 1フレーム目の送信の場合, KEY_ONへ
SCAF		
JNC	KEY_ON	
MOV	A, R07	⑮ 2フレーム目以降の送信 (連続押し) の場合, 送信中にキー・オフになったかを 判断します。連続押しの場合, KEY_48へ
SCAF		
JNC	KEY48	
;** Frame Space = 27ms **		
CALL	FS_27MS	⑯ ビット・データ送信中にキー・オフになった場合, フレーム・スペースの残り 27ms出力します。
HALT	#005H	⑰ HALTモード: タイマ・カウンタが00Hになると解除されます。
JMP	MAIN	初期設定処理, MAINへ
;		
KEY_ON:		
MOV	A, R17	⑱ 連続フラグ (R03) をセットします (このときR17=0FH)。
MOV	R03, A	
;** Frame Space = 27ms **		
CALL	FS_27MS	⑲ 27ms出力用サブルーチンをコールします。
JMP	TRNS	TRNSへ



```

;*****
;****          ****
;****          Transmit          ****
;****          Subroutine : Bit Out          ****
;****          ****
;*****
BITOUTOF:
    MOV    R1A, A                ① ワーク2 (R1A) に出力データを設定します。
    MOV    A, #0FH              ② 送信ビット・カウンタを1回に設定するためにアキュムレータを0FHに設定
                                します。
                                BITOUTへ
    JMP    BITOUT
    ;
BITOUTOC:
    MOV    R1A, A                ③ ワーク2 (R1A) に出力データを設定します。
    MOV    A, #0CH              ④ 送信ビット・カウンタを4回に設定するためにアキュムレータを0CHに設定
                                します。
BITOUT:
    MOV    ROA, A                [備考] 送信ビット・カウンタ (ROA) に0FH, 0CHのいずれかを設定します。

    MOV    A, R1A
    RL     A                    }----- ⑤ 出力データを左へ1ビット・シフトします。
    MOV    R1A, A
    JC     BIT_DAT1              ⑥ ビット・データが '1' か '0' を判断します。
; ** Bit Data = 0 **
    MOV    A, R08                }----- ⑦ データ・ポインタL (R08) にデータ0 (R08) を設定します。
    MOV    R00, A
    JMP    BIT_TRNS              BIT_TRNSへ
    ;
; ** Bit Data = 1 **
BIT_DAT1:
    MOV    A, R18                }----- ⑧ データ・ポインタL (R08) にデータ1 (R18) を設定します。
    MOV    R00, A
    MOV    A, R09
    INC    A                    }----- ⑨ データ1送信カウンタL (R09) をインクリメントします。
    MOV    R09, A
    JNC    BIT_TRNS              ⑩ データ1送信カウンタL (R09) がオーバーフローしなければ、
    MOV    A, R19                }----- ⑪ データ1送信カウンタH (R19) をインクリメントします。
    INC    A
    MOV    R19, A
BIT_TRNS:
    HALT   #005H                ⑫ HALTモード：タイマ・カウンタが00Hになると解除されます。
    MOV    T, @R0                ⑬ テーブル参照により、タイマ・カウンタに送信時間データを設定します。
    STTS   #0101B                ⑭ ステータス・フラグをクリアします。
    MOV    A, R00
    INC    A                    }----- ⑮ データ・ポインタL (R00) をインクリメントします。
    MOV    R00, A
    HALT   #005H                ⑯ HALTモード：タイマ・カウンタが00Hになると解除されます。

    MOV    T, @R0                ⑰ テーブル参照により、タイマ・カウンタに送信時間データを設定します。
    MOV    A, R07
    INC    A                    }----- ⑱ 10回連続してキー・オフが確定したかを判断します。
    JC     BIT_TRANS2              キー・オフ確定の場合、BIT_TRANS2へ
    STTS   #1110B                }----- ⑲ キー入力あり、かを判断します。
    JNF    BIT_TRANS1              キーが押し続けの場合、BIT_TRANS1へ
; ** キー入力あり **
    MOV    R7, #0F5H              ⑳ キー・オフ・チェック (10回) 中に1回でもキー入力があった場合は、キー・
                                オフ・チェック・カウンタ (R07) をイニシャライズ (=5H) します。
    JMP    BIT_TRANS2
    ;

```

```

; ** キー入力なし **
BIT_TRANS1:
    INC     A
    MOV     R07, A
BIT_TRANS2:
    STTS   #0101B
    MOV     A, R0A
    INC     A
    JNC    BITOUT

    RET
;
END

```

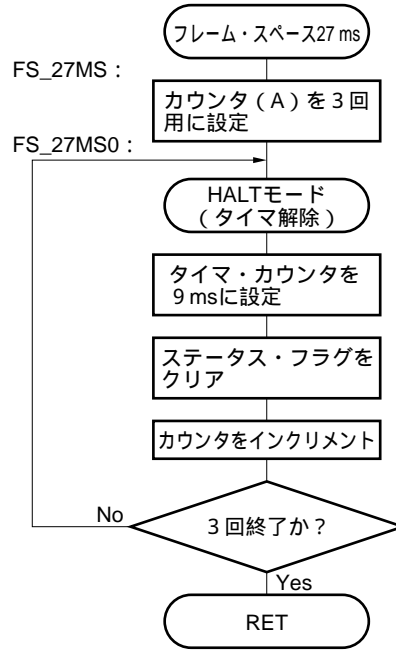
}----- ㉑ キー・オフ・チェック・カウンタをインクリメントします。

}----- ㉒ ステータス・フラグをクリアします。

}----- ㉓ 送信ビット・カウンタ (R0A) をインクリメントします。

}----- ㉔ 送信ビット数のカウントを終了したかを判断します。
 終了していない場合、BITOUTへ

}----- ㉕ 終了



```

;*****
;*****
;*****      FRAME SPACE = 27ms      *****  [備考] フレーム・スペースの残り27ms出力用サブルーチン
;*****      Subroutine : FS 27ms      *****
;*****
;*****
FS_27MS:
    MOV     A, #0DH                ① カウンタ (アキュムレータ) を3回 (=0DH) に設定します。
FS_27MS0:
    HALT   #005H                  ② HALTモード：タイマ・カウンタが00Hになると解除されます。
    MOV    T, #TIME9M             ③ タイマ・カウンタを9ms (=1FFH) に設定します。
    STTS   #0101B                 ④ ステータス・フラグをクリアします。
    INC    A                      ⑤ カウンタ (アキュムレータ) をインクリメントします。
    JNC    FS_27MS0              ⑥ 3回カウント終了したかを判断します。
                                       終了していない場合、FS_27MS0へ
    RET                               ⑦ 終了

```

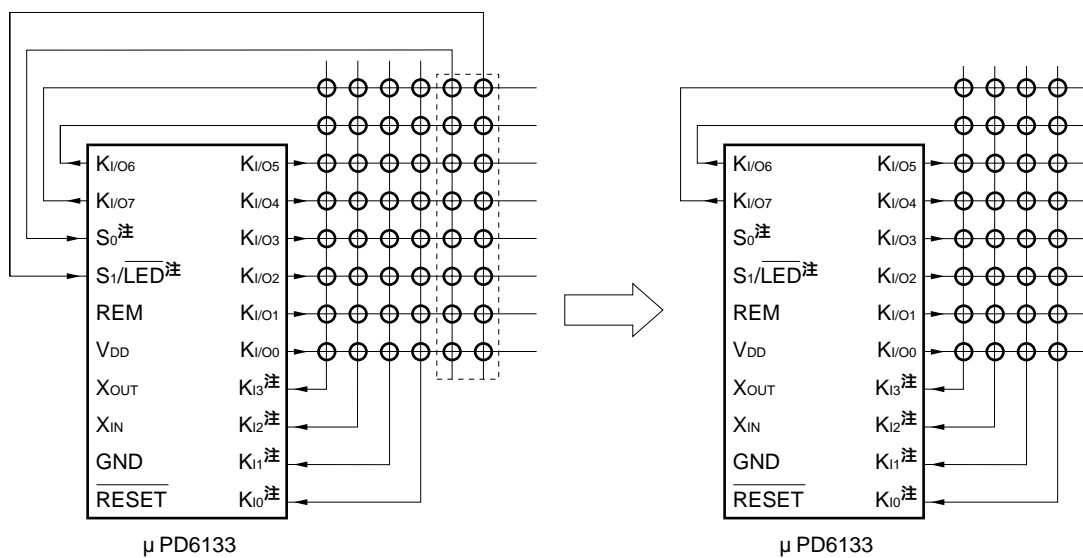

(メ モ)

第7章 プログラム変更時の注意点

キー・マトリクスおよびキー数の変更を行う際には、次の点に注意してください。

(1) 48キーから40キーまたは32キーに変更する場合

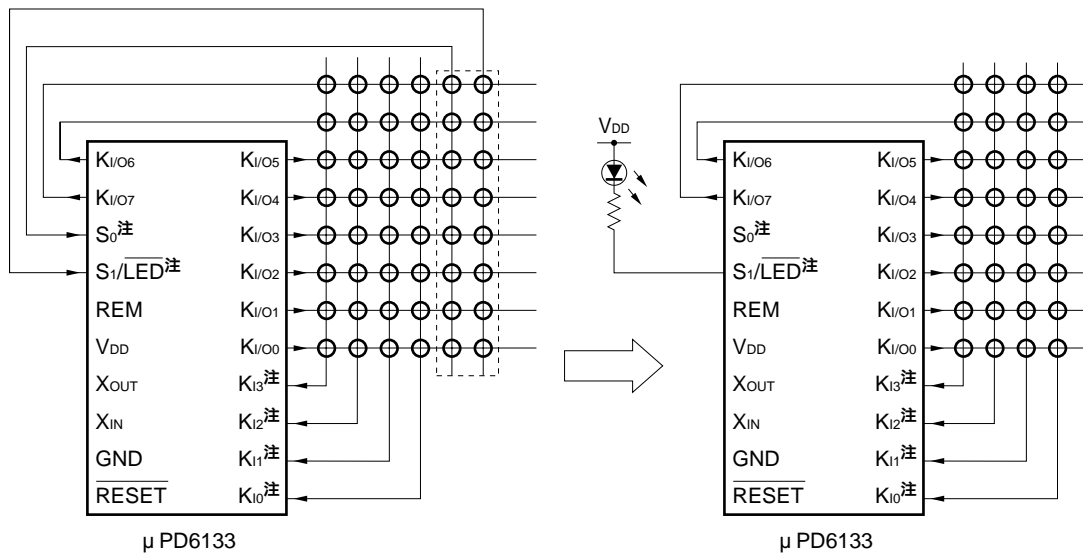
- ・次の図の破線で囲まれたキーを削除してください（そのとき、 S_0 および S_1/\overline{LED} 端子をオープンにすることによって、本プログラムの内容を変更せずにキー数を変更することができます）。



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

(2) (1)の変更に加えて、 S_1/\overline{LED} 端子をLEDとして使用する場合

- ・次の図の破線で囲まれたキーを削除してください。
- ・ S_0 端子はオープンにしてください。
- ・ S_1/\overline{LED} 端子は本プログラム内でコントロール・レジスタ1 (P4) の b_2 (S_1/\overline{LED} ポートの入出力モード設定用のビット) を常に出力モード (=1) に設定するように修正してください (μ PD6133 シリーズでは、 S_1/\overline{LED} 端子を出力モードにすることによって、内蔵プルダウン抵抗は自動的に切れます)。



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

第 8 章 プログラム・リスト

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:00 08/29/96 PAGE 01-002

SOURCE = MAIN48.ASM

```
E STNO LOC. OBJ.          M I SOURCE STATEMENT
1                          ;*****
2                          ;***
3                          ;***   Multi - Purpose Remote Control Transmitter System   ***
4                          ;***
5                          ;***   CPU           : uPD6133 Series           ***
6                          ;***   CPU Clock   : 455kHz             ***
7                          ;***   Trans. CODE: NEC-R Format (48Key)       ***
8                          ;***   Version    : 2.0                 ***
9                          ;***   Programmer : NEC IC Microcomputer Systems Corporation ***
10                         ;***
11                         ;***           Copyright(c) NEC Corporation 1995   ***
12                         ;***           Copyright(c) NIMS Corporation 1995   ***
13                         ;*****
14                         EJECT
```

第8章 プログラム・リスト

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:00 08/29/96 PAGE 01-003

SOURCE = MAIN48.ASM

```

E STNO LOC. OBJ.      M I SOURCE STATEMENT
15 001C              TIME05M      EQU      01CH              ;0.5ms (0.510ms)
16 01FF              DamyTime     EQU      512-1          ;9.00ms (= 17.582us * 512)
17
18                  ;##### P U B L I C #####
19                  PUBLIC      MAIN
20                  ;
21                  ;##### E X T E R N #####
22                  ;
23
24                  ;##### S T A R T #####
25
26                  ; Control Register (P3)      Reset:03H
27                  ;=====;
28                  ; D9 D8 ! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 ! ;
29                  ;          ! ! D.P.! D.P.! D.P.! TCTL! CARY! MOD1! MOD0! ;
30                  ; Test Mode! ! AD10! AD9 ! AD8 ! ! ! ! ! ;
31                  ;-----;
32                  ;          ! 0 ! * ! * ! * ! 1/1 ! ON ! fx,fx/8 ! 0 ;
33                  ; Set "0" !-----! ,fx/12(1/2)!----;
34                  ;          ! 0 ! * ! * ! * ! 1/2 ! OFF ! ,fx/12(1/3)! 1 ;
35                  ;=====;
36
37                  ; Control Register (P4)      Reset:26H
38                  ;=====;
39                  ; D9 D8 ! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 ! ;
40                  ;          ! ! ! KI !S0/S1! !S1/LED! KI/0! S0 ! ;
41                  ;          ! ! ! Pull! Pull! ! MODE ! MODE! MODE! ;
42                  ;-----;
43                  ;          ! 0 ! 0 ! OFF ! OFF ! 0 ! IN ! IN ! OFF ! 0 ;
44                  ; X !-----!-----!-----;
45                  ;          ! 0 ! 0 ! ON ! ON ! 0 ! OUT ! OUT ! IN ! 1 ;
46                  ;=====;
47
48                  ;#####
49                  ; Control Register Initialize
50                  ;#####
51 MAIN:
52 0000 E6F8 EFEF      OUT      P0,#0FFH          ;KI/0 All High
53 0002 E6FC E3E3      OUT      P4,#033H
54 0004 E6FB E1E3      OUT      P3,#013H          ;Set Data Pointer (P3:D8,D9)
55 0006 E6FF F7FF      MOV      T,#DamyTime
56 0008 E3F1 E0E5      STTS     #0101B          ;Clear Status Flag
57
58                  ;#####
59                  ; RAM Initialize Routine
60                  ;#####
61 000A E6E1 E0E0      MOV      R1,#000H          ;Final Key Data (R1) = 00H
62 000C E6E3 E0E0      MOV      R3,#000H          ;K29 ON Flag (R13) , Continuance Flag (R03) = 00H
63
64 000E E2F1 E0E8      HALT     #008H            ;STOP mode (KI = High)
65 0010 E0E0          NOP                      ;No operation command
66 0011 E6FF E0E7      MOV      T,TIME05M        ;Set Timer : 0.5ms
67                  ;
68 END

```

SOURCE = KEY48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
1                          ;*****
2                          ;****          ****
3                          ;****      uPD6133 Series Key Check      ****
4                          ;****      Trans. CODE: NEC-R Format (48Key) ****
5                          ;****          ****
6                          ;*****
7      01FF              TIME9M EQU 1FFH          ;9.00ms (9.002ms)
8      0050              CUSTM1 EQU 050H          ;Custom Code = 0AH
9      00AF              CUSTM2 EQU 0AFH          ;Custom Code' = F5H
10
11                          ;##### P U B L I C #####
12                          PUBLIC      KEY48
13                          ;
14
15                          ;##### E X T E R N #####
16                          EXTRN      MAIN          ;MAIN Routine
17                          ;
18
19                          ;##### S T A R T #####
20      KEY48:
21      0013 E6E0 E0EC          MOV     R0,#00CH          ;Chattering Counter (R00) = 3 Times
22                          ;*****
23                          ;          ON Chattering
24                          ;*****
25      ON_CHAT:
26      0015 E3F1 E0E5          STTS   #0101B          ;Clear Status Flag
27      0017 E6F8 EF8F          OUT   P0,#0FFH          ;K1/O All High
28      0019 E6FC E3E3          OUT   P4,#033H
29      001B E2F1 E0E5          HALT  #005H          ;HALT mode (Timer = 00H)
30      001D E6FF F7FF          MOV   T,#TIME9M          ;Timer = 9.00ms
31      001F E6E9 E0E0          MOV   R9,#000H          ;Comparative Key Data (R9) = 00H
32      0021 E6E6 E0E8          MOV   R6,#008H          ;Key Flag (R16) = 00H , Key Scan Counter (R06) = 08H
33      0023 E3F1 E0EE          STTS  #1110B
34      0025 EFF1 E0E0          JNF   MAIN              ; With input Key (K1 or S0 or S1) ?
35                          ;          else MAIN: (Without input Key)
36                          ;          then (With input Key)
36      0027 E6E8 E0E1          MOV   R8,#001H          ;Key Scan Data (R8) Initialize : 001H
37
38                          ;*****
39                          ;          Key Scan
40                          ;*****
41      KEYS_OUT:
42      0029 FFE8          MOV   A,R08
43      002A E5F8          OUT  P00,A              ;Key Scan Data L (P00) Output
44      002B FEE8          MOV  A,R18
45      002C E4F8          OUT  P10,A              ;Key Scan Data H (P10) Output
46                          ;+++++
47                          ;++      100us Wait      ++
48                          ;+++++
49      002D E6E2 EEE1          MOV  R2,#0E1H          ;Set Immediate Data : R2
50      002F E6EB ECE3          MOV  RB,#0C3H          ;Set Immediate Data : RB
51      0031 E0E0          NOP
52      0032 E0E0          NOP
53      0033 E0E0          NOP
54      0034 E0E0          NOP
55                          ;*****

```

SOURCE = KEY48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56                      ;           Key Return Check
57                      ;*****
58 0035 E3F1 E0EB          STTS  #1011B          ;With input Key (K1 or S0 or S1) ?
59 0037 EFF1 E9E3          JNF   NO_KEYS          ;   else NO_KEYS: (Without input Key)
60                          ;   then          (With input Key)
61 0039 E3F1 E0E3          STTS  #0011B          ;With input K1 (P11) ?
62 003B EFF1 E4E9          JNF   SOS1CALC        ;   else SOS1CALC: (With input K1)
63                          ;   then          (Without input K1)
64                      ;*****
65                      ;           KI Data Calculate
66                      ;*****
67 003D FFF9              IN    A,P01          ;With input K1 , Does it have S0 or S1 ?
68 003E FCF3              RL    A              ;With input S0 and S1 , It judges " MULTIPLE KEYS ERROR " .
69 003F ECF1 E0E0          JC    MAIN          ; and then goes to " MAIN: " !
70 0041 FCF3              RL    A
71 0042 ECF1 E0E0          JC    MAIN
72
73 0044 FEEB              MOV   A,R1B          ;R1B = 0CH
74 0045 E5EA              MOV   R0A,A          ;Key Return Check Counter (R0A) = 4 Times (=0CH)
75 0046 FEF9              IN    A,P11
76 0047 E8F1 E4EE          JMP   KEY_RETO
77                      ;
78
79                      ;*****
80                      ;           S0,S1 Data Calculate
81                      ;*****
82 SOS1CALC:
83 0049 E6E9 ECE0          MOV   R9,#0C0H
84 004B FEE2              MOV   A,R12          ;R12 = 0EH
85 004C E5EA              MOV   R0A,A          ;Set Key Return Check Counter (R0A) = 2 Times (=0EH)
86 004D FFF9              IN    A,P01
87
88 KEY_RETO:
89 004E E4E0              MOV   R10,A          ;Work (R10) = P11 or P01
90 004F E3F1 E0E5          STTS  #0101B          ;Clear Status Flag
91 KEY_RET:
92 0051 FEE0              MOV   A,R10
93 0052 FEF3              RLZ   A
94 0053 E4E0              MOV   R10,A
95 0054 EDF1 E8EE          JNC   KRC_INC        ;With input Key ?
96                      ;** With input Key **          ;   else KRC_INC: (Without input Key)
97 0056 FEE6              MOV   A,R16          ;   then          (With input Key)
98 0057 FAF3              SCAF          ;Key Flag (R16) = 0FH ?
99 0058 EDF1 E6E9          JNC   KEY_CALC        ;   else KEY_CALC: (Except for 0FH)
100 005A FEE3              MOV   A,R13          ;   then          (= 0FH)
101 005B FAF3              SCAF          ;K29 ON Flag (R13) = 0FH ?
102 005C EDF1 E0E0          JNC   MAIN          ;   else MAIN:    (= Except for 0FH)
103                          ;   then          (= 0FH)
104 005E FFF1 F4E0          MOV   A,#0100B
105 0060 F4E9              XRL   A,R19
106 0061 FAF3              SCAF          ;Comparative Key Data = K29 (R19-0BH) ?
107 0062 EDF1 E0E0          JNC   MAIN          ;   then MAIN:    (Key = Except for K29)
108 0064 FFE9              MOV   A,R09          ;   else          (Key = K29)
109 0065 F5EB              XRL   A,ROB          ;ROB = 03H
110 0066 FAF3              SCAF          ;Comparative Key Data = K29 (R09-0CH) ?

```

SOURCE = KEY48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111 0067 EDF1 E0E0                JNC     MAIN          ;   then DBL_K_CHK: (Key = Except for K29)
112
113                               ;*****
114                               ;   Key Data Calculate
115                               ;*****
116 KEY_CALC:
117 0069 FFE6                MOV     A,R0B          ;R0B = 03H
118 006A F4E9                XRL   A,R19
119 006B FAF3                SCAP
120 006C EDF1 E7E5                JNC   K1_CALC          ;Comparative Key Data H (R19) = 0CH ?
121                               ;** S0 or S1 data Calculation **;   then (S0 or S1 Data)
122 006E FFE6                MOV     A,R06          ;Acc <-- Key Scan Counter (R06)
123 006F FCF3                RL     A
124 0070 FAE2                ANL   A,R12           ;R12 = 1110B
125 0071 E5E9                MOV   R09,A
126 0072 FFE2                MOV   A,R02           ;R02 = 01H
127 0073 E8F1 E8E1                JMP   KL_CALC
128
129                               ;** K1 data Calculation **
130 K1_CALC:
131 0075 FFE6                MOV   A,R06           ;Acc <-- Key Scan Counter(R06)
132 0076 FCF3                RL   A
133 0077 FCF3                RL   A                ;Left shift (2 Times)
134 0078 E5E9                MOV   R09,A          ;Comparative Key Data L (R09) <-- Acc
135 0079 FBE2                ANL  A,R02           ;Acc AND 0001B (R02=01H)
136 007A FDF1 FAED                ORL  A,#1010B        ;Acc OR 1010B
137 007C E4E9                MOV   R19,A          ;Comparative Key Data H (R19) <-- Acc
138 007D FFE9                MOV   A,R09          ;Acc <-- Comparative Key Data L (R09)
139 007E FAE8                ANL  A,R1B           ;Acc AND 1100B (R1B=0CH)
140 007F E5E9                MOV   R09,A          ;Comparative Key Data L (R09) <-- Acc
141 0080 FFE8                MOV   A,R0B          ;Acc <-- 0011B (R0B=03H)
142
143 KL_CALC:
144 0081 FBFA                ANL  A,R0A           ;Acc AND Key Return Check Counter (R0A)
145 0082 FDE9                ORL  A,R09           ;Acc OR Comparative Key Data L (R09)
146 0083 E5E9                MOV   R09,A          ;Comparative Key Data L (R09) <-- Acc
147
148                               ;** Double Key Check **
149 0084 FEE6                MOV   A,R16
150 0085 FAF3                SCAP
151 0086 EDF1 E8EB                JNC   SINGLE_K        ;Key Flag (R16) = 0FH ?
152 0088 FFF1 F9E0                MOV   A,#09H          ;   else SINGLE_K (Single Key)
153 008A E4E9                MOV   R19,A          ;   then (Double Key)
154 SINGLE_K:
155 008B FFF1 FFE0                MOV   A,#0FH
156 008D E4E6                MOV   R16,A
157
158 KRC_INC:
159 008E FFEA                MOV   A,R0A
160 008F F4F3                INC   A                ;Key Return Check Counter (R0A) increment
161 0090 E5EA                MOV   R0A,A          ;End of Key Return Check ?
162 0091 EDF1 E5E1                JNC   KEY_RET          ;   else KEY_RET: (Continuous)
163                               ;   then (End)
164
165 NO_KEYS:

```


SOURCE = KEY48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
166 0093 E3F1 E0E5          STTS  #0101B          ;Clear Status Flag
167 0095 FFE6              MOV   A, R06
168 0096 F4F3              INC   A                ;Key Scan Counter (R06) increment
169 0097 E5E6              MOV   R06, A          ;End of Key Scan number of 8 Time ?
170 0098 ECF1 EAEG          JC    KEY_DATA        ; then KEY_DATA: (End)
171 009A FEE8              MOV   A, R18          ; else (Continuous)
172 009B FCF3              RL   A                ;Key Scan Data H (R18) Shift
173 009C E4E8              MOV   R18, A
174 009D FFE8              MOV   A, R08
175 009E FCF3              RL   A
176 009F E5E8              MOV   R08, A          ;Key Scan Data L (R08) Shift --> CY ?
177 00A0 EDF1 E2E9          JNC  KEYS_OUT         ; else KEYS_OUT: (Key Scan Data a low rank)
178 00A2 E6E8 E1E0          MOV   R8, #010H       ; then (Key Scan Data a high rank)
179 00A4 E8F1 E2E9          JMP  KEYS_OUT
180
181
182
183          ;*****
184          ; Transmit Key Data = ON Chattering Key Data ?
185          ;*****
186          KEY_DATA:
187          OUT   P0, #OFFH          ;K1/O All High
188          MOV   R8, #0F0H          ;Set Immediate Data : R8
189          MOV   A, R16
190          SCAF
191          JNC  MAIN                ;Key Flag (R16) = 0FH ?
192          MOV   A, R00             ; else MAIN: (= Except for 0FH)
193          XRL  A, R0B             ; then (= 0FH)
194          SCAF                    ;R0B = #0011B
195          JC    KEYCHECK           ;Chattering counter (R00) = 1 Times (=0DH) ?
196          MOV   A, R14             ; then KEYCHECK: (= 0DH)
197          XRL  A, R18             ; else (= 0E-0FH)
198          XRL  A, R19             ;R18 = 0FH
199          SCAF                    ;Acc = 0FH
200          JNC  MAIN                ;Key Data H (R14) = Comparative Key Data H (R19)
201          XRL  A, R04             ; else MAIN: (Unmatch)
202          XRL  A, R09             ; then (Match)
203          SCAF                    ;Acc = 0FH
204          JNC  MAIN                ;Key Data L (R04) = Comparative Key Data L (R09)
205          MOV   A, R19             ; else MAIN: (Unmatch)
206          XRL  A, R14             ; then (Match)
207          MOV   A, R19
208          MOV   R14, A            ;Key Data H (R14) <-- Comparative Key Data H (R19)
209          MOV   A, R09
210          MOV   R04, A            ;Key Data L (R04) <-- Comparative Key Data L (R09)
211          MOV   A, R00
212          INC  A                ;Chattering Counter(R00) increment
213          MOV   R00, A
214          SCAF
215          JNC  ON_CHAT            ;End of Chattering Routine of 3 Time ?
216          ;** Key High Check **   ; else ON_CHAT: (= 0DH)
217          XRL  A, R11             ; then (= 0E-0FH)
218          XRL  A, R14             ;Acc = 0FH
219          SCAF
220          JNC  KDAT_SET           ;Final Key Data H (R11) = Key Data H (R14) ?
                ; else KDAT_SET: (Unmatch)

```

SOURCE = KEY48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
221                                ;** Key Low Check ** ;      then      (Match)
222 00CD F5E1              XRL   A,R01      ;Acc = 0FH
223 00CE F5E4              XRL   A,R04
224 00CF FAF3              SCAF                      ;Final Key Data L(R01) = Key Data L (R04) ?
225 00D0 ECF1 EDE8        JC    K29CHECK      ;      then K29CHECK: (Match)
226                                ;      else      (Unmatch)
                                KDAT_SET:
227 00D2 FEE4              MOV   A,R14
228 00D3 E4E1              MOV   R11,A          ;Final Key Data H (R11) <-- Key Data H (R14)
229 00D4 FFE4              MOV   A,R04
230 00D5 E5E1              MOV   R01,A          ;Final Key Data L (R01) <-- Key Data L (R04)
231 00D6 FFE8              MOV   A,R08
232 00D7 E5E3              MOV   R03,A          ;Continuance Flag = 00H (R08=00H)
233
234                                K29CHECK:
235 00D8 FFF1 F4E0        MOV   A,#0100B
236 00DA F4E1              XRL   A,R11
237 00DB FAF3              SCAF                      ;Final Key Data = K29 (R11=0BH) ?
238 00DC EDF1 EEE3        JNC   DBL_K_CHK      ;      then DBL_K_CHK: (Key = Except for K29)
239 00DE FFE1              MOV   A,R01          ;      else      (Key = K29)
240 00DF F5EB              XRL   A,R0B          ;R0B = 03H
241 00E0 FAF3              SCAF                      ;Final Key Data = K29 (R01=0CH) ?
242 00E1 ECF1 EEEA        JC    K29_FLG        ;      then K29_FLG: (Key = Except for K29)
243                                ;      else      (Key = K29)
244                                DBL_K_CHK:
245 00E3 FFF1 F6E0        MOV   A,#06H
246 00E5 F4E1              XRL   A,R11
247 00E6 FAF3              SCAF                      ;Final Key Data = K29+ K30 or K31 or K32 (R11=09H) ?
248 00E7 ECF1 EEEB        JC    DATA_SET      ;      then DATA_SET: (Key = Except for Double Key)
249 00E9 FFE8              MOV   A,R08
250                                K29_FLG:
251 00EA E4E3              MOV   R13,A          ;K29 ON Flag = 00H (R08=00H)
252                                ;*****
253                                ;      Transmit Data Set
254                                ;*****
255                                DATA_SET:
256                                ;** Custom Code **
257 00EB E6E4 E5E0        MOV   R4,#CUSTM1      ;Custom Code(R4) <-- Custom Code
258
259                                ;** Custom Code' **
260 00ED E6E5 EAEF        MOV   R5,#CUSTM2      ;Custom Code'(R5) <-- Custom Code'
261
262                                ;** Data Code **
263 00EF FEE1              MOV   A,R11
264 00F0 E4E0              MOV   R10,A          ;Data Pointer H(R10) <-- Final Key Data H(R11)
265 00F1 FFE1              MOV   A,R01
266 00F2 E5E0              MOV   R00,A          ;Data Pointer L(R00) <-- Final Key Data L(R01)
267 00F3 E7E6              MOV   R6,@R0
268
269                                END

```

TOTAL ERRORS = 0
TOTAL WARNINGS = 0

END OF LIST

SOURCE = TRANS48.ASM

```

E STNO LOC. OBJ.      M I SOURCE STATEMENT
1
2
3
4
5
6
7 01FF              TIME9M      EQU      1FFH          ;9.00ms (9.002ms)
8
9
10
11
12
13
14
15
16
17
18
19
20 00F4 E6E7 EFE5    MOV      R7,#0F5H          ;R17 : Set Immediate Data (=0FH)
21
22 00F6 E6E9 EDEF    MOV      R9,#0DFH          ;Key OFF Check Counter (R07) = 10 Times (=05H)
23
24
25
26 00F8 E6E0 EDE0    MOV      R0,#0D0H          ;Data Pointer H (R10) = 0DH
27 00FA E6E8 EDE8    MOV      R8,#008H          ;Set Table Address L
28 00FC FEE8          MOV      A,R18             ;Output Data (Work) <-- 00H (R18)
29 00FD E6F2 E8F1 E5FF CALL     BITOUTOF
30
31
32
33
34 0100 E6E8 ECEA    MOV      R8,#0CAH          ;Set Table Address L
35 0102 FEE4          MOV      A,R14             ;Output Data (Work) <-- Custom Code H (R14)
36 0103 E6F2 E8F1 E6F4 CALL     BITOUTOC
37
38 0106 FFE4          MOV      A,R04             ;Output Data (Work) <-- Custom Code L (R04)
39 0107 E6F2 E8F1 E6F4 CALL     BITOUTOC
40
41
42
43
44 010A FEE5          MOV      A,R15             ;Output Data (Work) <-- Custom Code' H (R15)
45 010B E6F2 E8F1 E6F4 CALL     BITOUTOC
46
47 010E FFE5          MOV      A,R05             ;Output Data (Work) <-- Custom Code' L (R05)
48 010F E6F2 E8F1 E6F4 CALL     BITOUTOC
49
50
51
52
53 0112 FEE6          MOV      A,R16             ;Output Data (Work) <-- Data Code H (R16)
54 0113 E6F2 E8F1 E6F4 CALL     BITOUTOC
55

```

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 03-003

SOURCE = TRANS48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56 0116 FFE6              MOV    A,R06                ;Output Data (Work) <-- Data Code L (R06)
57 0117 E6F2 E8F1 E6F4    CALL  BITOUTOC
58
59                        ;*****
60                        ;          Data Code'
61                        ;*****
62 011A FEE6              MOV    A,R16                ;Output Data (Work) <-- Data Code H (R16)
63 011B F5F1 FFE0        XRL   A,#0FH
64 011D E6F2 E8F1 E6F4    CALL  BITOUTOC
65
66 0120 FFE6              MOV    A,R06                ;Output Data (Work) <-- Data Code L (R06)
67 0121 F5F1 FFE0        XRL   A,#0FH
68 0123 E6F2 E8F1 E6F4    CALL  BITOUTOC
69
70                        ;*****
71                        ;          Stop Bit
72                        ;*****
73 0126 E6E8 E0EE        MOV    R8,#00EH            ;Set Table Address L
74 0128 FEE8              MOV    A,R18                ;Output Data (Work) <-- 00H (R18)
75 0129 E6F2 E8F1 E5FF    CALL  BITOUTOF
76
77                        ;*****
78                        ;          Frame Space
79                        ;*****
80 012C FEE9              MOV    A,R19
81 012D FAF3              SCAP
82 012E ECF1 E3FE        JC     T9_UNDER            ;Frame Space > 9.00ms ?
83                        ;           then T9_UNDER: (More than 9.00ms)
84 0130 E2F1 E0E5        HALT  #005H                ;           else (Less than 9.00ms)
85 0132 E6FF F7FF        MOV    T,#TIME9M          ;HALT mode (Timer = 00H)
86 0134 E3F1 E0E5        STTS  #0101B              ;Clear Status Flag
87
88 0136 FFE9              MOV    A,R09
89 0137 FCF3              RL    A                    ;Frame Space > 18.00ms ?
90 0138 ECF1 E3FE        JC     T9_UNDER            ;           then T9_UNDER: (Less than 18.00ms)
91                        ;           else (More than 18.00ms)
92 013A E2F1 E0E5        HALT  #005H                ;HALT mode (Timer = 00H)
93 013C E6FF F7FF        MOV    T,#TIME9M
94
95 T9_UNDER:
96 013E E3F1 E0E5        STTS  #0101B              ;Clear Status Flag
97 0140 FFE9              MOV    A,R09
98 0141 FBF1 F7E0        ANL   A,#0111B
99 0143 E5E0              MOV    R00,A
100 0144 E2F1 E0E5        HALT  #005H                ;HALT mode (Timer = 00H)
101 0146 E7FF              MOV    T,#R0
102 0147 E3F1 E0E5        STTS  #0101B              ;Clear Status Flag
103
104 0149 FFE3              MOV    A,R03
105 014A FAF3              SCAP                        ;Continuously pressed key ?
106 014B EDF1 E5F8        JNC   KEY_ON              ;           else KEY_ON: (1st. Frame)
107 014D FFE7              MOV    A,R07                ;           then (Since 2nd. Frame)
108 014E FAF3              SCAP                        ;Without input Key ?
109 014F EDF1 E1E3        JNC   KEY48                ;           then KEY48: (With input Key)
110                        ;           else (Without input Key)
                        ;** Frame Space = 27ms **

```

SOURCE = TRANS48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111 0151 EXTRN                CALL   FS_27MS
112 0154 E2F1 E0E5            HALT   #005H          ;HALT mode (Timer = 00H)
113 0156 E8F1 E0E0            JMP    MAIN
114                            ;
115
116 0158 FEE7                MOV    A,R17
117 0159 E5E3                MOV    R03,A          ;Continuance Flag (R03) = 0FH (R17=0FH)
118                            ;** Frame Space = 27ms **
119 015A EXTRN                CALL   FS_27MS
120 015D E8F1 EFEA            JMP    TRNS
121                            ;
122
123
124                            ;*****
125                            ;*****
126                            ;***** Transmit *****
127                            ;***** Subroutine : Bit Out *****
128                            ;*****
129                            ;*****
130
131 015F E4EA                MOV    R1A,A          ;Set Output Data (1 Bit Output)
132 0160 FFF1 FFE0            MOV    A,#0FH         ;Send Bit Counter = 1 Times (=0FH)
133 0162 E8F1 E6F7            JMP    BITOUT
134                            ;
135
136 0164 E4EA                MOV    R1A,A          ;Set Output Data (4 Bit Output)
137 0165 FFF1 FCE0            MOV    A,#0CH         ;Send Bit Counter = 4 Times (=0CH)
138
139
140 0167 E5EA                MOV    R0A,A          ;Send Bit Counter (R0A) = 0CH or 0EH or 0FH
141
142 0168 FEEA                MOV    A,R1A
143 0169 FCF3                RL    A
144 016A E4EA                MOV    R1A,A          ;Bit Data = 1 ?
145 016B ECF1 E7F1            JC    BIT_DAT1        ; then BIT_DAT1: (Bit Data = 1)
146                            ;** Bit Data = 0 ** ; else (Bit Data = 0)
147 016D FFE8                MOV    A,R08          ;Set Data Pointer L (R00) <-- Data0 (R08)
148 016E E5E0                MOV    R00,A
149 016F E8F1 E7FB            JMP    BIT_TRNS
150                            ;
151                            ;** Bit Data = 1 **;
152
153 0171 FEE8                MOV    A,R18          ;Set Data Pointer L(R00) <-- Data1 (R18)
154 0172 E5E0                MOV    R00,A
155 0173 FFE9                MOV    A,R09
156 0174 F4F3                INC    A              ;Data 1 Transmit Counter L (R09) increment
157 0175 E5E9                MOV    R09,A          ;Data 1 Transmit Counter L (R09) = Overflow ?
158 0176 EDF1 E7FB            JNC   BIT_TRNS        ; else BIT_TRNS: (Data 1 Transmit Counter L <= 0FH)
159 0178 FEE9                MOV    A,R19          ; then (Data 1 Transmit Counter L > 0FH)
160 0179 F4F3                INC    A              ;Data 1 Transmit Counter H(R19) increment
161 017A E4E9                MOV    R19,A
162
163 017B E2F1 E0E5            HALT   #005H          ;HALT mode (Timer = 00H)
164 017D E7FF                MOV    T,#R0
165 017E E3F1 E0E5            STTS  #0101B         ;Clear Status Flag

```

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 03-005

SOURCE = TRANS48.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
166 0180 FFE0             MOV    A,R00
167 0181 F4F3             INC    A                      ;Data Pointer L (R00) increment
168 0182 E5E0             MOV    R00,A
169 0183 E2F1 E0E5       HALT   #005H                  ;HALT mode (Timer = 00H)
170 0185 E7FF             MOV    T,#R0
171
172 0186 FFE7             MOV    A,R07
173 0187 F4F3             INC    A                      ;Key Off Check Counter (R07) increment
174 0188 ECF1 E9F3       JC     BIT_TRANS2            ;Key Off Check Counter (R07) = End of 10 Times ?
175                                     ;   then BIT_TRANS2: (End of 10 Times)
176                                     ;   else           (Less than 10 Times)
177 018A E3F1 E0EE       STTS   #1110B                ;With input key ?
178 018C EFF1 E9F2       JNF    BIT_TRANS1            ;   else BIT_TRANS1: (With input)
179                                     ;** With input key ** ;   then           (Without input)
180 018E E6E7 EFE5       MOV    R7,#0F5H              ;Key off Check Counter (R07) = 05H
181 0190 E8F1 E9F3       JMP    BIT_TRANS2
182                                     ;
183                                     ;** Without input key **
184 BIT_TRANS1:
185 0192 E5E7             MOV    R07,A
186
187 BIT_TRANS2:
188 0193 E3F1 E0E5       STTS   #0101B                ;Clear Status Flag
189 0195 FFEA             MOV    A,ROA
190 0196 F4F3             INC    A                      ;Send Bit Counter (ROA) increment
191 0197 EDF1 E6F7       JNC   BITOUT
192 0199 E8F2             RET
193                                     ;
194 END

```

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

SOURCE = DTABLE48.TBL

```

E STNO LOC. OBJ.      M I SOURCE STATEMENT
1
2
3          ;*****
4          ;*****      uPD6133 Series Data Table      *****
5          ;*****      Trans. CODE: NEC-R Format (48Key) *****
6          ;*****
7 01FF          TIME9M      EQU      1FFH          ;9.00ms(9.002ms)
8
9          ;#####      P U B L I C      #####
10         PUBLIC      DATACODE
11         PUBLIC      FS_27MS
12         ;
13
14         DATACODE:
15         ;*****
16         ;*****      Key Data      *****
17         ;*****      Double Key      *****
18         ;*****
19 019D          ORG      19DH
20         ;** K29 + K30 **
21 019D E0E9          DW      009H          ;K29 + K30 : Key Data = 0DH
22         ;** K29 + K31 **
23 019E E8E9          DW      089H          ;K29 + K31 : Key Data = 0EH
24         ;** K29 + K32 **
25 019F E4E9          DW      049H          ;K29 + K32 : Key Data = 0FH
26         ;
27
28         ;*****
29         ;*****      Key Data      *****
30         ;*****      Single Key      *****
31         ;*****
32 01A0          ORG      1A0H
33         ;** K 1 - K16 **
34 01A0 E0E0          DW      000H          ;K 1 : Key Data = 20H
35 01A1 E8E0          DW      080H          ;K 2 : Key Data = 21H
36 01A2 E4E0          DW      040H          ;K 3 : Key Data = 22H
37 01A3 ECE0          DW      0C0H          ;K 4 : Key Data = 23H
38 01A4 E2E0          DW      020H          ;K 5 : Key Data = 24H
39 01A5 EAE0          DW      0A0H          ;K 6 : Key Data = 25H
40 01A6 E6E0          DW      060H          ;K 7 : Key Data = 26H
41 01A7 EEE0          DW      0E0H          ;K 8 : Key Data = 27H
42 01A8 E1E0          DW      010H          ;K 9 : Key Data = 28H
43 01A9 E9E0          DW      090H          ;K10 : Key Data = 29H
44 01AA E5E0          DW      050H          ;K11 : Key Data = 2AH
45 01AB EDE0          DW      0D0H          ;K12 : Key Data = 2BH
46 01AC E3E0          DW      030H          ;K13 : Key Data = 2CH
47 01AD EBEO          DW      0B0H          ;K14 : Key Data = 2DH
48 01AE E7E0          DW      070H          ;K15 : Key Data = 2EH
49 01AF EFE0          DW      0F0H          ;K16 : Key Data = 2FH
50         ;
51
52         ;** K17 - K32 **
53 01B0 E0E8          DW      008H          ;K17 : Key Data = 30H
54 01B1 E8E8          DW      088H          ;K18 : Key Data = 31H
55 01B2 E4E8          DW      048H          ;K19 : Key Data = 32H

```

第 8 章 プログラム・リスト

AS6133 VI.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 04-003

SOURCE = DTABLE48.TBL

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56 01B3 ECE8             DW 0C8H          ;K20 : Key Data = 33H
57 01B4 EZE8             DW 028H          ;K21 : Key Data = 34H
58 01B5 EAE8             DW 0A8H          ;K22 : Key Data = 35H
59 01B6 EGE8             DW 068H          ;K23 : Key Data = 36H
60 01B7 EEE8             DW 0E8H          ;K24 : Key Data = 37H
61 01B8 E1E8             DW 018H          ;K25 : Key Data = 38H
62 01B9 E9E8             DW 098H          ;K26 : Key Data = 39H
63 01BA E5E8             DW 058H          ;K27 : Key Data = 3AH
64 01BB EDE8             DW 0D8H          ;K28 : Key Data = 3BH
65 01BC E3E8             DW 038H          ;K29 : Key Data = 3CH
66 01BD EBE8             DW 0B8H          ;K30 : Key Data = 3DH
67 01BE E7E8             DW 078H          ;K31 : Key Data = 3EH
68 01BF EFE8             DW 0F8H          ;K32 : Key Data = 3FH
69                          ;
70                          ;** K33 - K48 **
71 01C0 E0E4             DW 004H          ;K33 : Key Data = 40H
72 01C1 E8E4             DW 084H          ;K34 : Key Data = 41H
73 01C2 E4E4             DW 044H          ;K35 : Key Data = 42H
74 01C3 ECE4             DW 0C4H          ;K36 : Key Data = 43H
75 01C4 E2E4             DW 024H          ;K37 : Key Data = 44H
76 01C5 EAE4             DW 0A4H          ;K38 : Key Data = 45H
77 01C6 EGE4             DW 064H          ;K39 : Key Data = 46H
78 01C7 EEE4             DW 0E4H          ;K40 : Key Data = 47H
79 01C8 E1E4             DW 014H          ;K41 : Key Data = 48H
80 01C9 E9E4             DW 094H          ;K42 : Key Data = 49H
81 01CA E5E4             DW 054H          ;K43 : Key Data = 4AH
82 01CB EDE4             DW 0D4H          ;K44 : Key Data = 4BH
83 01CC E3E4             DW 034H          ;K45 : Key Data = 4CH
84 01CD EBE4             DW 0B4H          ;K46 : Key Data = 4DH
85 01CE E7E4             DW 074H          ;K47 : Key Data = 4EH
86 01CF EFE4             DW 0F4H          ;K48 : Key Data = 4FH
87                          ;
88
89
90                          ;*****
91                          ;****   Frame Space   ****
92                          ;****   Time Data     ****
93                          ;ORG 1D0H *****
93 01D0 E7ED             DT 1F4H          ;Counter:000 = 8.815ms (8.809ms)
94 01D1 E6ED             DT 1B4H          ;Counter:001 = 7.690ms (7.683ms)
95 01D2 E5ED             DT 174H          ;Counter:010 = 6.565ms (6.558ms)
96 01D3 E4ED             DT 134H          ;Counter:011 = 5.440ms (5.433ms)
97 01D4 E3ED             DT 0F4H          ;Counter:100 = 4.315ms (4.308ms)
98 01D5 E2ED             DT 0B4H          ;Counter:101 = 3.190ms (3.182ms)
99 01D6 E1ED             DT 074H          ;Counter:110 = 2.065ms (2.057ms)
100 01D7 E0ED            DT 034H          ;Counter:111 = 0.940ms (0.932ms)
101                          ;
102
103                          ;*****
104                          ;****   Time Data     ****
105                          ;ORG 1D8H *****
106                          ;** Leader Code **
107 01D8 FFFF             DT 3FFH          ;Carrier ON : 9.000ms (9.002ms)
108 01D9 F3FF             DT 0FFH          ;Carrier OFF : 4.500ms (4.501ms)
109                          ;** Bit data 0 **
110 01DA F8F7             DT 21FH          ;Carrier ON : 0.560ms (0.563ms)

```


AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 04-004

SOURCE = DTABLE48.TBL

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111 01DB F0F7             DT    01FH          ;Carrier OFF : 0.565ms (0.563ms)
112                      ;** Bit data 1 **
113 01DC F8F7             DT    21FH          ;Carrier ON  : 0.560ms (0.563ms)
114 01DD F1F7             DT    05FH          ;Carrier OFF : 1.690ms (1.687ms)
115                      ;** Stop Bit **
116 01DE F8F7             DT    21FH          ;Carrier ON  : 0.560ms (0.563ms)
117 01DF E2FA             DT    0A9H          ;Carrier OFF : 3.000ms (2.989ms)
118                      ;
119
120
121
122                      ;*****
123                      ;*****
124                      ;**** Frame Space = 27ms ****
125                      ;**** Subroutine : FS 27ms ****
126                      ;*****
127                      ;*****
128
129                      FS_27MS:
130 01E0 FFF1 FDE0             MOV    A,#0DH          ;Counter(Acc) = 3 Times (=0DH)
131                      FS_27MS0:
132 01E2 E2F1 E0E5             HALT   #005H          ;HALT mode (Timer = 00H)
133 01E4 E6FF F7FF             MOV    T,#TIME9M      ;Timer = 9.00ms
134 01E6 E3F1 E0E5             STTS   #0101B         ;Clear Status Flag
135 01E8 F4F3                 INC    A               ;More than 27ms ?
136 01E9 EDF1 EEF2             JNC    FS_27MS0       ; else FS_27MS0: (Frame Space < 27ms)
137 01EB E8F2                 RET                    ; then RET (Frame Space = 27ms)
138                      ;
139                      END

```

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 21:25:49 06/03/96 PAGE 05-002

SOURCE = OPTION.ASM

E	STNO	LOC.	OBJ.	M	I	SOURCE STATEMENT
	1					*****
	2					***** Set Option *****
	3					*****
	4					OPTION
	5	0000	0001			USEPOC
	6					ENDOP
	7					END

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

〔メ モ〕

第3編 80キー対応のプログラム

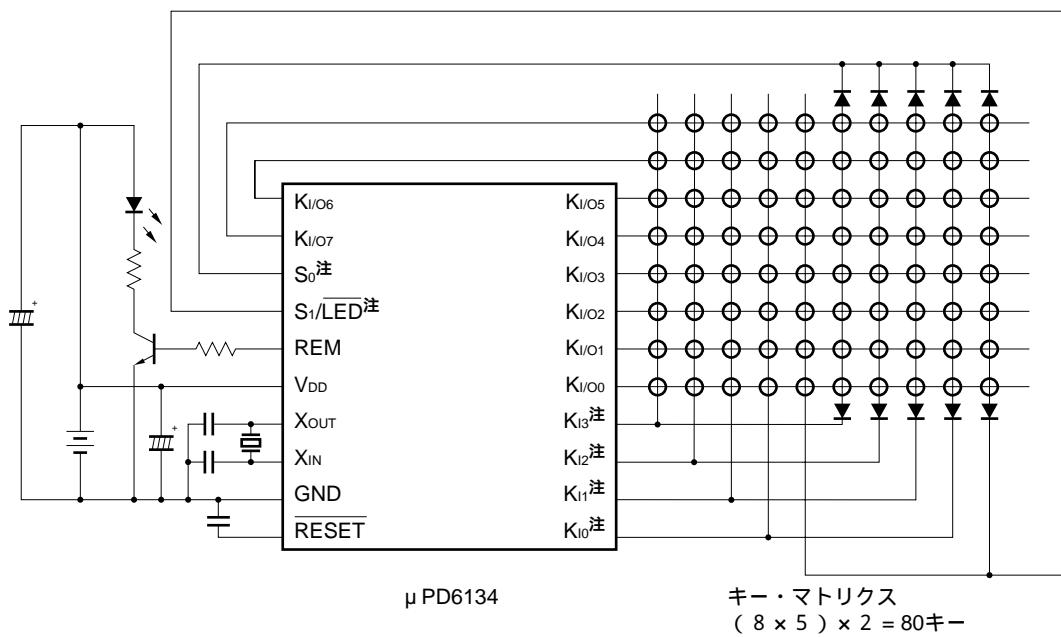
〔メ モ〕

第1章 ハードウェア構成

1.1 応用回路例

図1-1に応用回路例を示します。

図1-1 応用回路例



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

K1, K1/O, S0, S1/LED端子で80キーのキー・マトリクスを構成しています。送信コードはREM端子からキャリア付きで出力されます。

各端子の機能を表1-1に示します。

表1-1 端子の機能

端子名	入出力	機能
K1/00-K1/07	出力	キー・ソース (アクティブ・ハイ)
K10-K13	入力	キー・リターン (アクティブ・ハイ)
S0	入力	キー拡張
S1/LED	入力	キー・リターン (アクティブ・ハイ)
REM	出力	赤外線リモコン信号 (キャリア付き)

1.2 キー・マトリクス

図1-2にキー・マトリクスを示します。

各キーのKn (n = 1 ~ 80) という記号は、キーの位置を示しています。

図1-2 キー・マトリクス

K1/07	K29	K30	K31	K32	K40	K69	K70	K71	K72	K80
K1/06	K25	K26	K27	K28	K39	K65	K66	K67	K68	K79
K1/05	K21	K22	K23	K24	K38	K61	K62	K63	K64	K78
K1/04	K17	K18	K19	K20	K37	K57	K58	K59	K60	K77
K1/03	K13	K14	K15	K16	K36	K53	K54	K55	K56	K76
K1/02	K9	K10	K11	K12	K35	K49	K50	K51	K52	K75
K1/01	K5	K6	K7	K8	K34	K45	K46	K47	K48	K74
K1/00	K1	K2	K3	K4	K33	K41	K42	K43	K44	K73
	K13	K12	K11	K10	S1	K13	K12	K11	K10	S1
	S0 = 0					S0 = 1 (拡張)				

第2章 送信波形

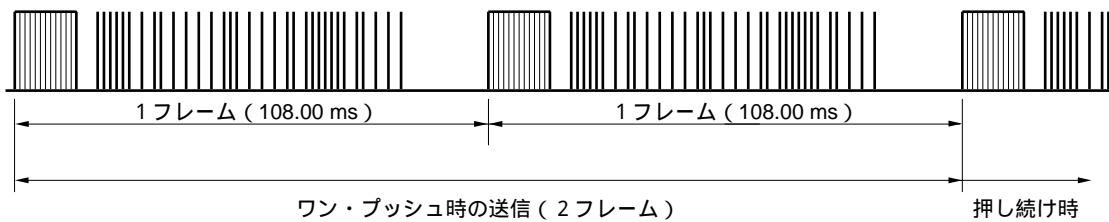
REM端子から出力される送信波形は、NEC連続送信（以下、NEC-Rと記す）フォーマットです。
出力データ・コードの内容については、第4章 出力コードを参照してください。

2.1 NEC-Rフォーマット

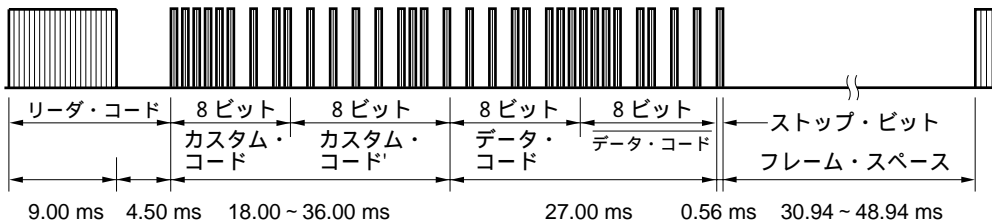
NEC-Rフォーマットを図2 - 1に示します。

図2 - 1 NEC-Rフォーマット

・送信波形

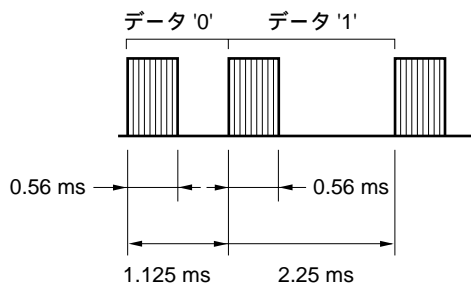


・1フレームの送信波形

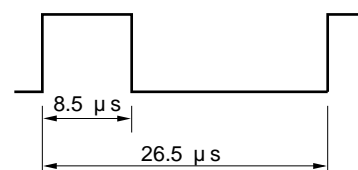


・2フレーム目以降の送信波形... 1フレーム目の送信波形と同様

・ビット・データ形式



・キャリアの波形



$f_x = 455 \text{ kHz}$
キャリア周波数: 38 kHz
デューティ比: 1/3

〔メ モ〕

第3章 タイミング・チャート

3.1 キー入力-REM出力タイミング・チャート

オン・チャタリング除去処理は、9.00 msごとにキー入力をチェックし、3回続けてオンと判定されたとき、キーがオンになったと判断します。また、オン・チャタリング除去処理ではキーの状態変化（キーのオン/オフ、キーの押し続け）をチェックしています。

次にオフ・チャタリング除去処理の方法を示します。

ビット・データのロウ・レベル出力中にキー・オフのチェックを行います。

1フレーム（108 ms）の送信中、REM端子からロウ・レベルを出力するタイミング（34回）で、10回キー入力をチェックして、

（1）10回連続でキー・オフと判定された場合

...キーがオフになったと判断します。

（2）10回のチェック中に1回でもキー・オンと判定された場合

...キー・オンと判定された時点で、チェック回数のカウンタをクリアし、さらに10回キー入力をチェックします。

（3）（1）と（2）のチェックでキーがオフになったと判断されなかった場合

...キーが押し続けられている（キーがオンである）と判断します。

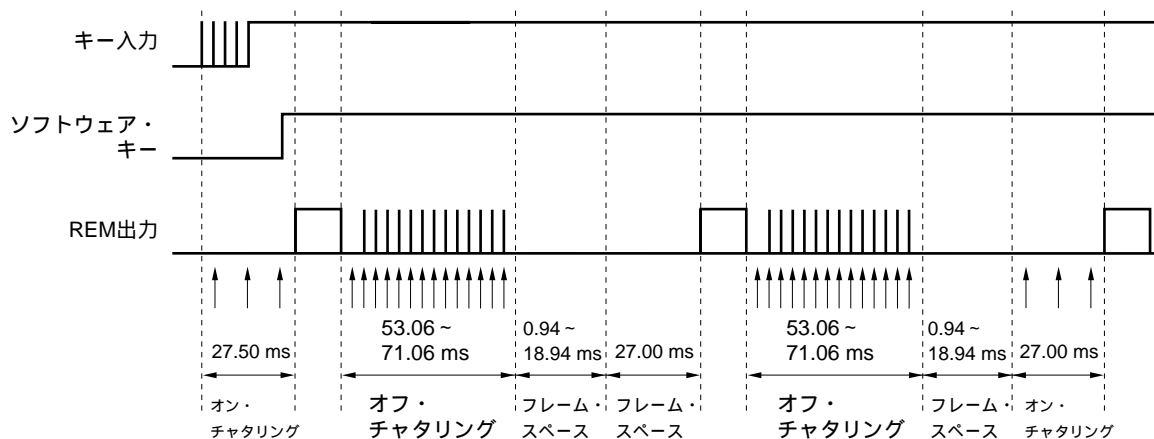
1フレーム目送信中にキーがオフになったと判断した場合でも、最低2フレームの送信を行ってから初期設定処理を行います。

3.1.1 キーの押し続け時のタイミング

図3 - 1にキーを押し続けた場合のタイミング・チャートを示します。

オン・チャタリング除去処理の詳細は、6.2.1（1）**チャタリング除去処理**を参考にしてください。

図3 - 1 キーを押し続けたときのタイミング・チャート



- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード, ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため, フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか, またはキー・オフが確定した場合, オン・チャタリング除去処理は行われずに, フレーム・スペース送信が行われます (= 27.00 ms)。

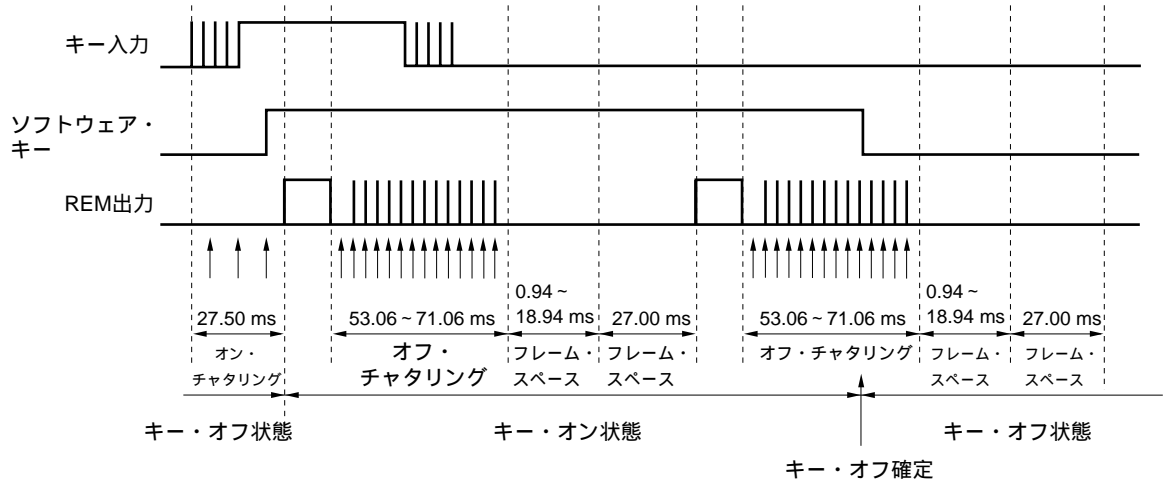
3.1.2 キー・オフ時のタイミング

図3 - 2にキーが送信途中で離された場合のタイミング・チャートを示します。

図3-2 キー・オフ時のタイミング・チャート (1/3)

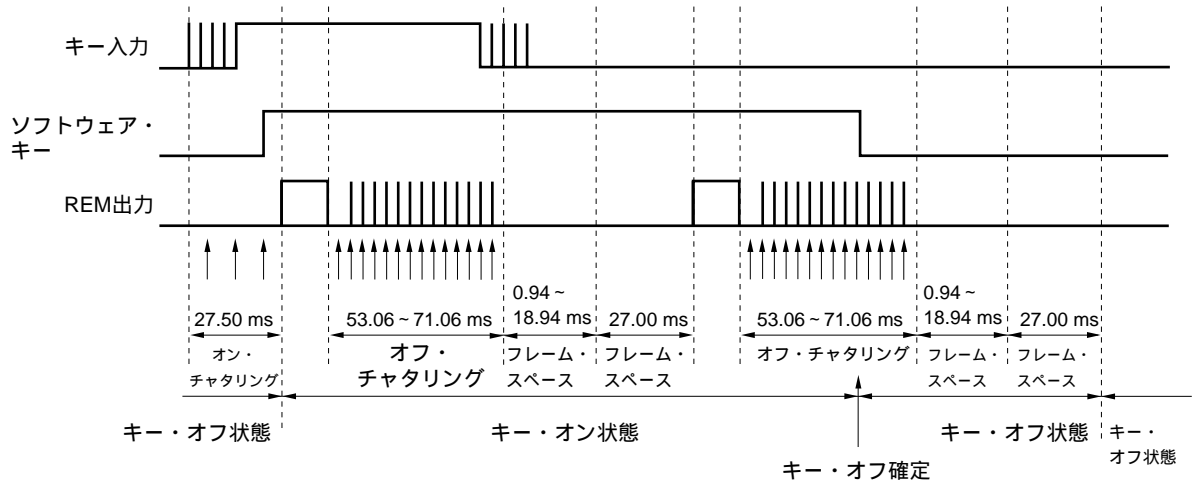
(1) 1フレーム目の送信中にキーが離された場合 (1フレーム目の送信中に、10回連続でキー・オフと判定された場合)

...キー・オフは、2フレーム目の送信中に確定します。



(2) 1フレーム目の送信中にキーが離された場合 (1フレーム目の送信中に、10回連続でキー・オフと判断されなかった場合)

...キー・オフは、2フレーム目の送信中に確定します。

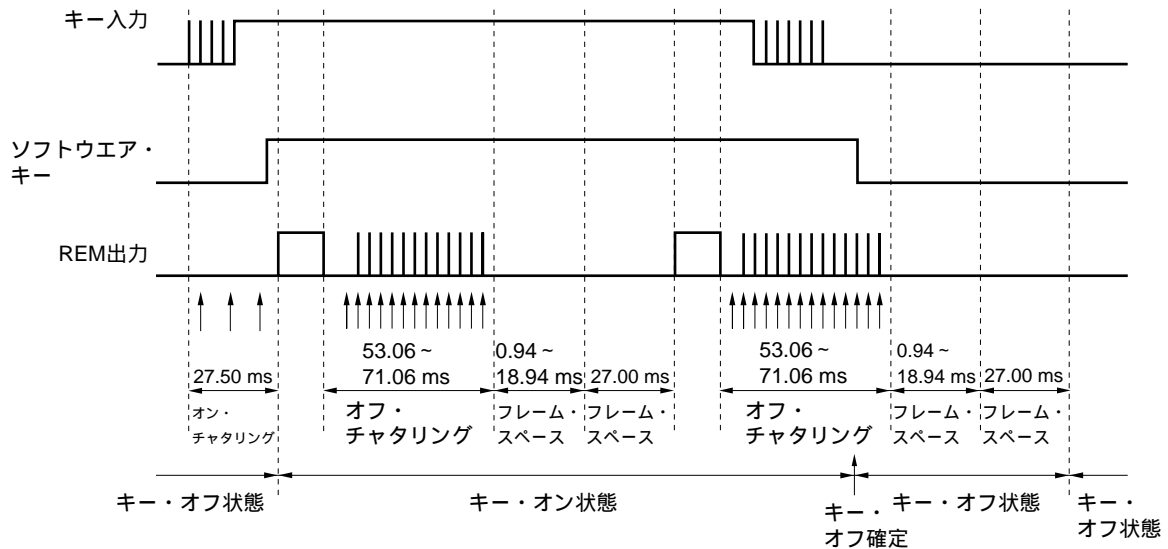


- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード、ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われずに、フレーム・スペース送信が行われます (= 27.00 ms)。

図3 - 2 キー・オフ時のタイミング・チャート (2/3)

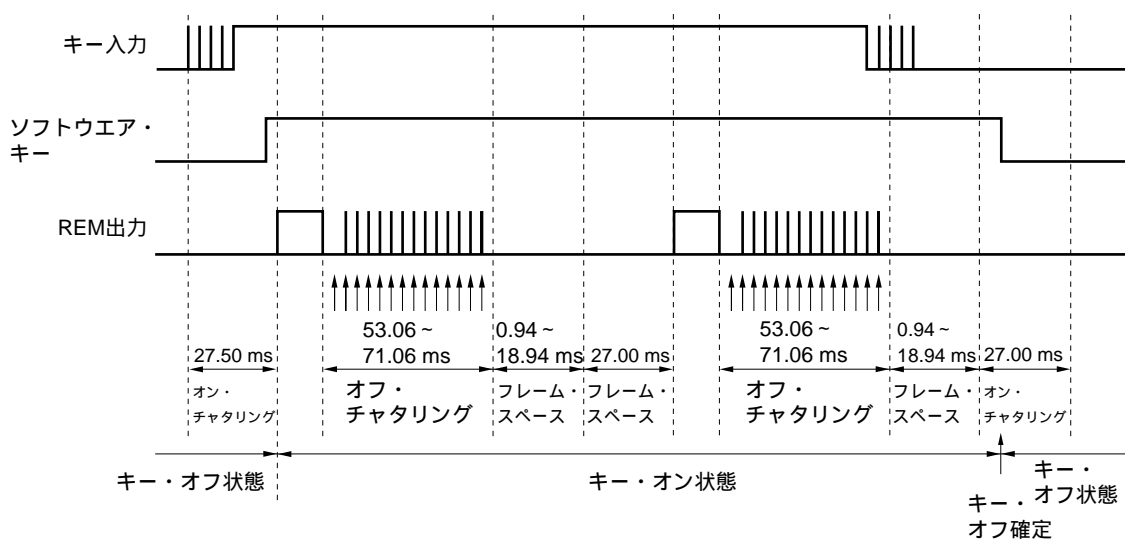
(3) 2フレーム目の送信中にキーが離された場合 (2フレーム目の送信中に、10回連続でキー・オフと判定された場合)

...キー・オフは、2フレーム目の送信中に確定します。



(4) 2フレーム目の送信中にキーが離された場合 (2フレーム目の送信中に、10回連続でキー・オフと判断されなかった場合)

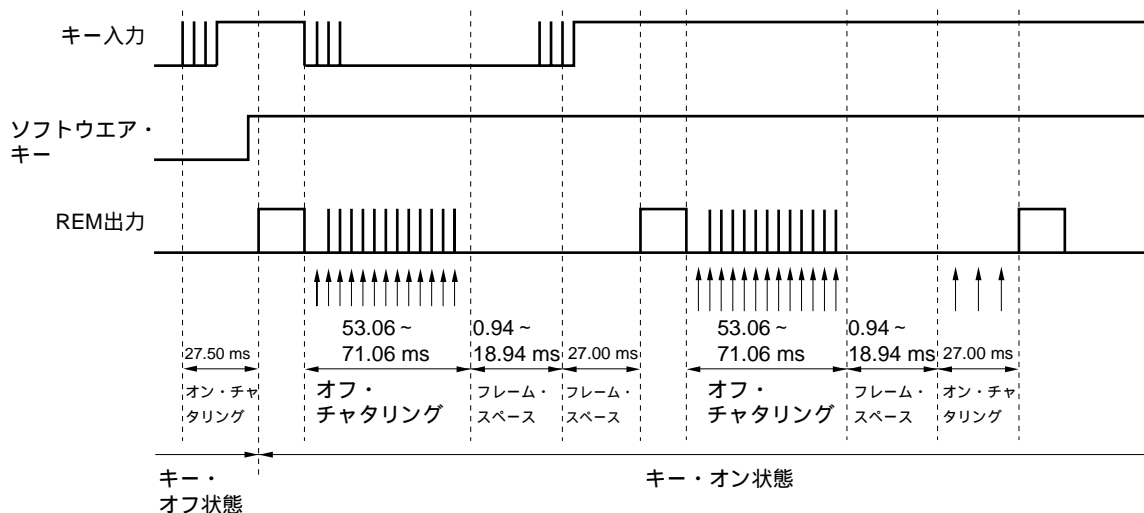
...キー・オフは、キーのオン・チャタリングで確定します。



- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード、ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われずに、フレーム・スペース送信が行われます (= 27.00 ms)。

図3 - 2 キー・オフ時のタイミング・チャート (3/3)

(5) 1フレーム目の送信途中でキーが離され、その後キーが押された場合
 ... 2フレーム目でキー・オンのため、キー・オフは確定しません。



- キー入力 : 実際のキー動作 (手操作でキーを押す / 離す) です。
- ソフトウェア・キー : ソフトウェア上でのキー動作 (プログラム上でのキー入力判断) です。
- REM出力 : REM端子より出力される送信波形です。
- オン・チャタリング : オン・チャタリング除去処理です。9.00 msごとに3回キー入力をチェックします。
- オフ・チャタリング : オフ・チャタリング除去処理です。リーダー・コード, ストップ・ビットを含めたビット間 (34ビット分) のキーの入力をチェックします。
- フレーム・スペース : カスタム・コードの送信時間により変化するため、フレーム・スペースで1フレームの時間が108.00 msになるように調整します。
- フレーム・スペース : 1フレーム目であるか、またはキー・オフが確定した場合、オン・チャタリング除去処理は行われずに、フレーム・スペース送信が行われます (= 27.00 ms)。

3.2 キー動作のタイミング・チャート

3.2.1 キー確定までの出力パターン

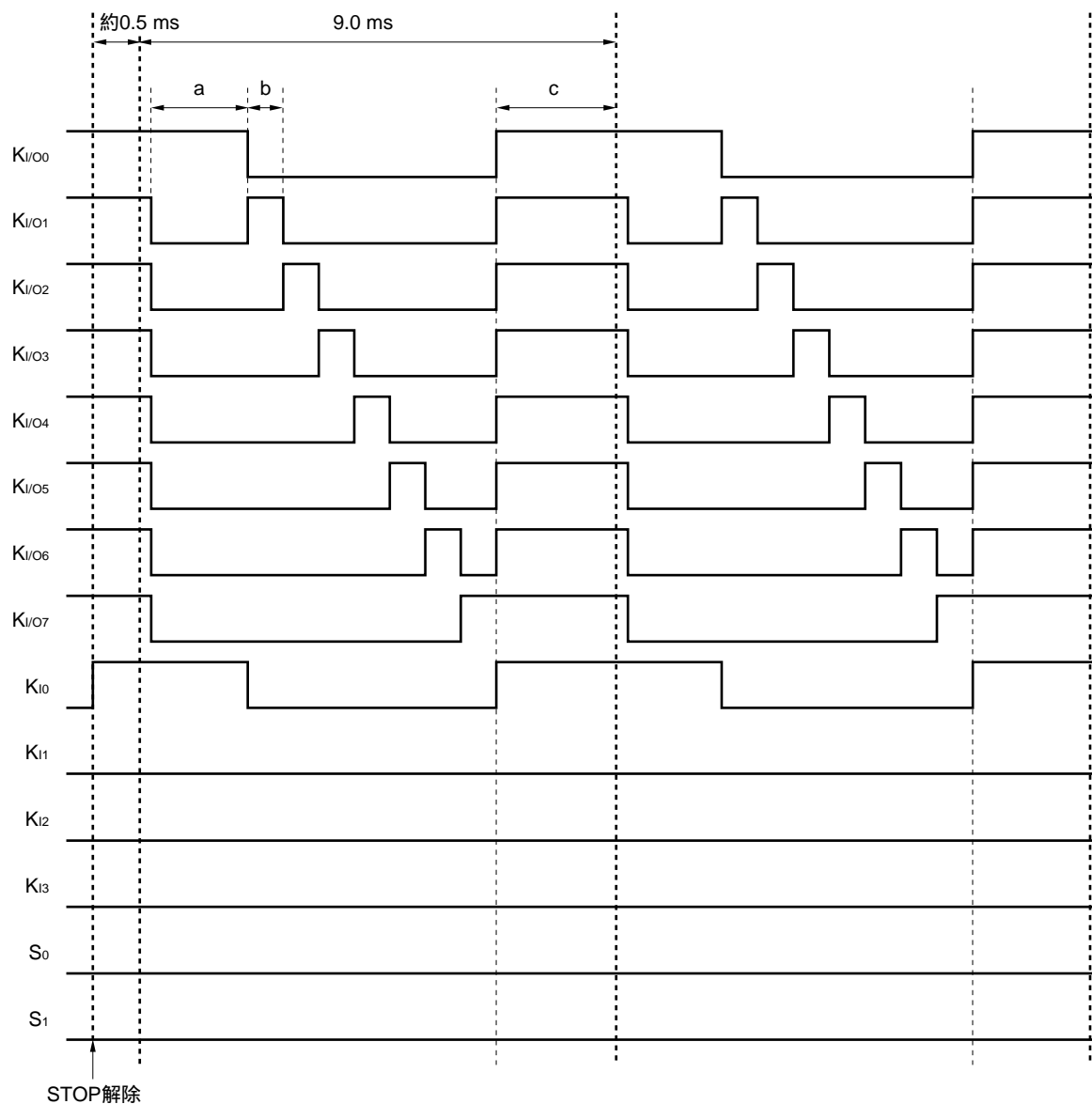
キー確定までには、オン・チャタリング除去処理 (9.00 ms x 3回 = 27.00 ms) を行い、キーが3回同じ状態であればキーを確定します。

キー入力の確認は、キー・ソース出力端子をハイ・レベルに設定してから100 μs以上 (fx = 455 kHz時, 6命令以上) 経過後に判断します。

キー確定までの詳細を、図3 - 3に示します。

図3 - 3 キー確定までの出力パターン (1/6)

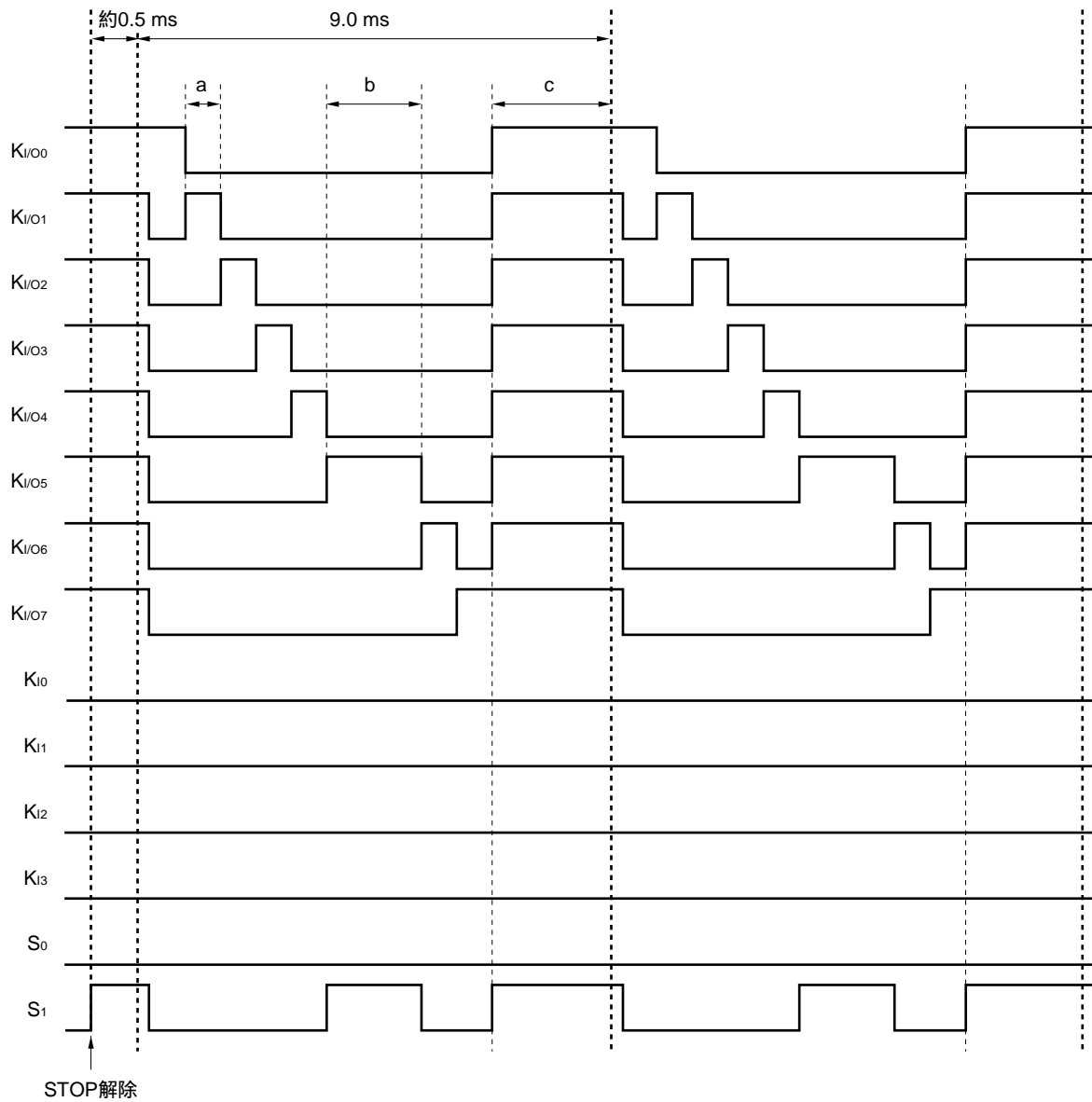
(1) K4が押された場合



- a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
実行時間は約1.32 ms ~ 2.02 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。
- c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。
実行時間は約4.18 ms ~ 4.88 msかかります。

図3 - 3 キー確定までの出力パターン (2/6)

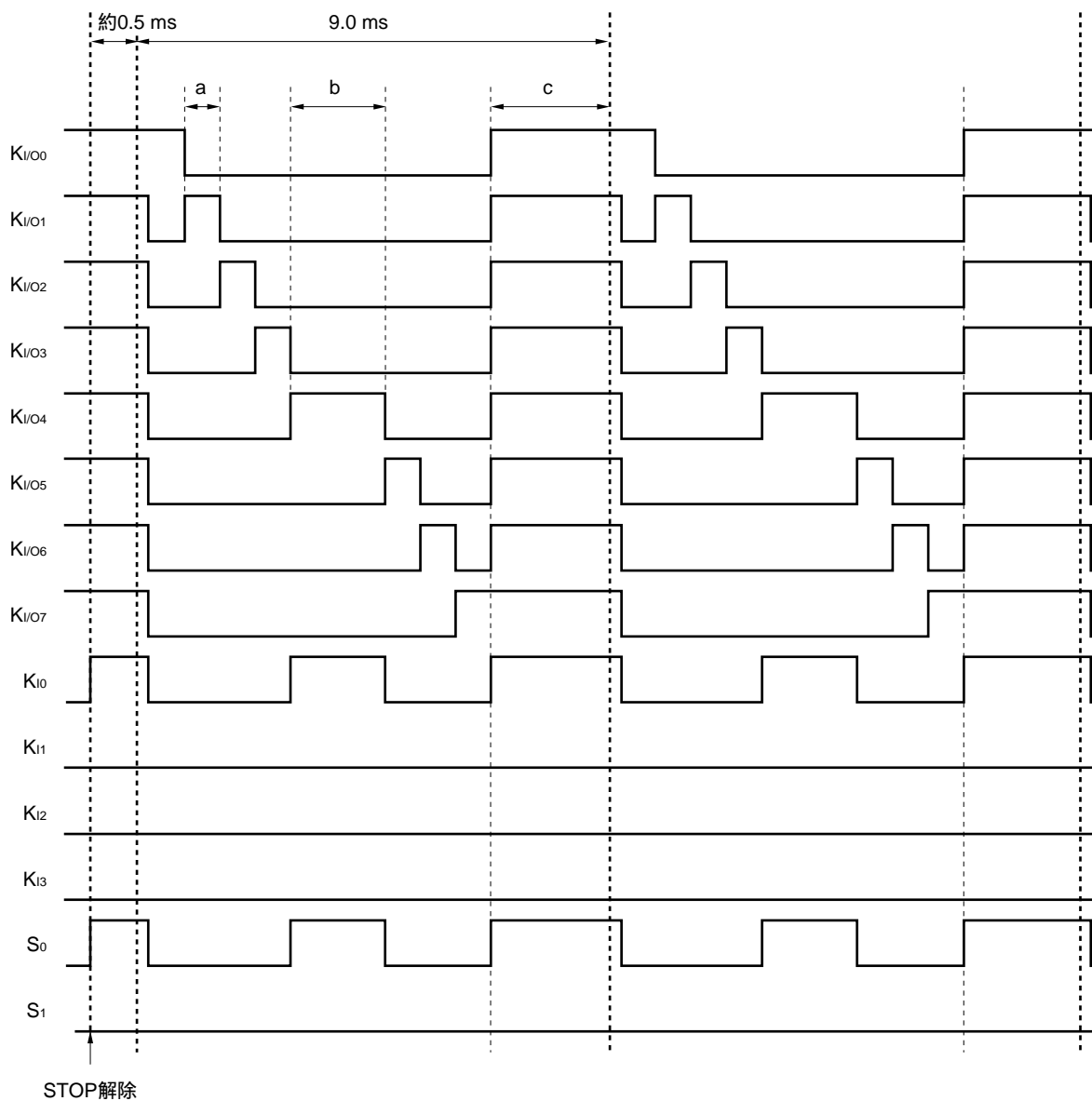
(2) K38が押された場合



- a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
実行時間は約1.32 ms ~ 2.02 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。
- c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。
実行時間は約4.18 ms ~ 4.88 msかかります。

図3 - 3 キー確定までの出力パターン (3/6)

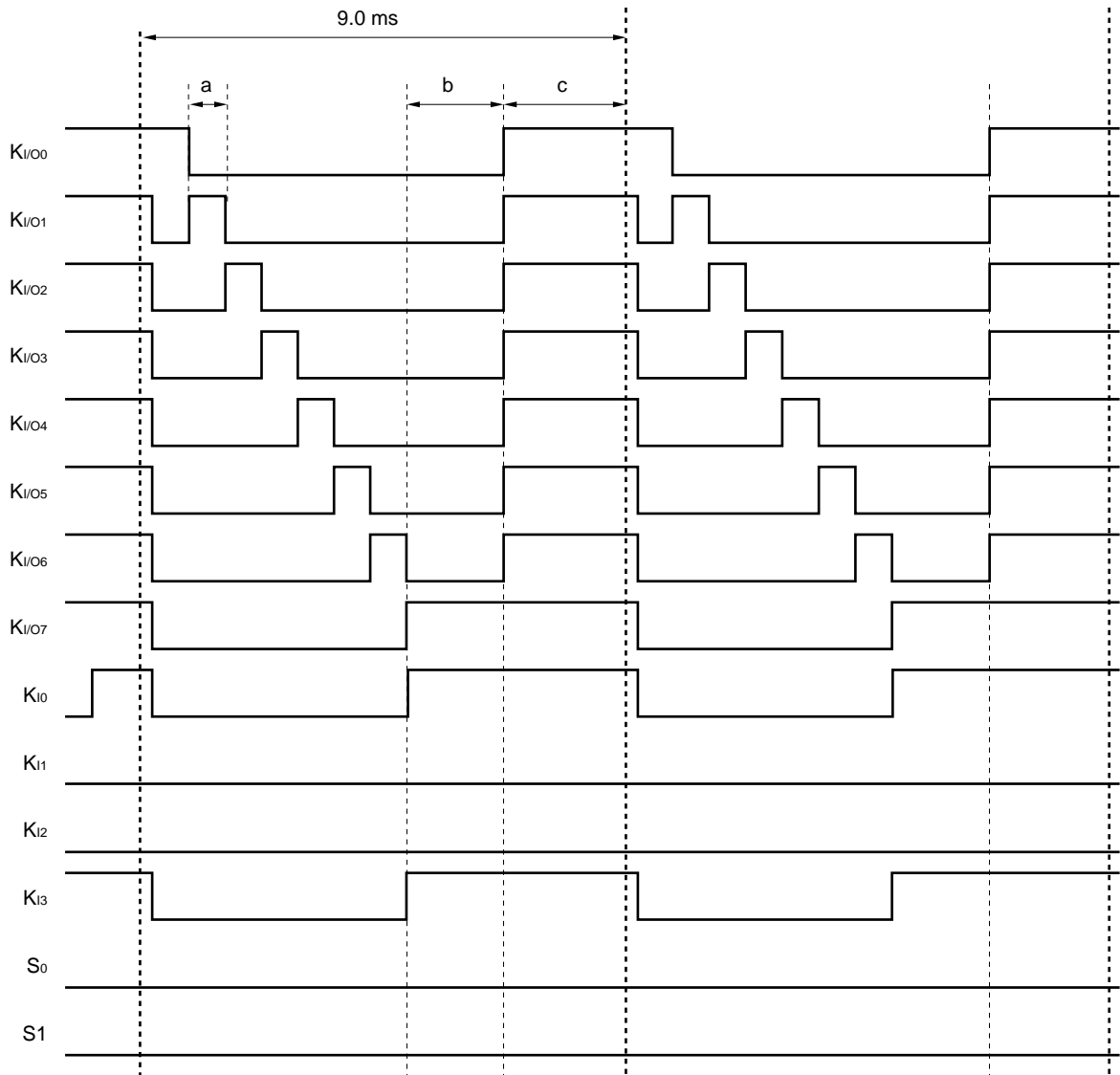
(3) K60が押された場合



- a : キーがオンであると判断した場合, キー・リターンをチェックし, キー・データを算出します。
実行時間は約1.32 ms ~ 2.02 msです。
- b : キー入力のチェックを行います。
実行時間は約0.41 msです。
- c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。
実行時間は約4.18 ms ~ 4.88 msかかります。

図3 - 3 キー確定までの出力パターン (4/6)

(4) K29との有効な二重押しキー (K32) が押された場合 (K29 + K32)



a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。

実行時間は約1.32 ms ~ 2.02 msです。

b : キー入力のチェックを行います。

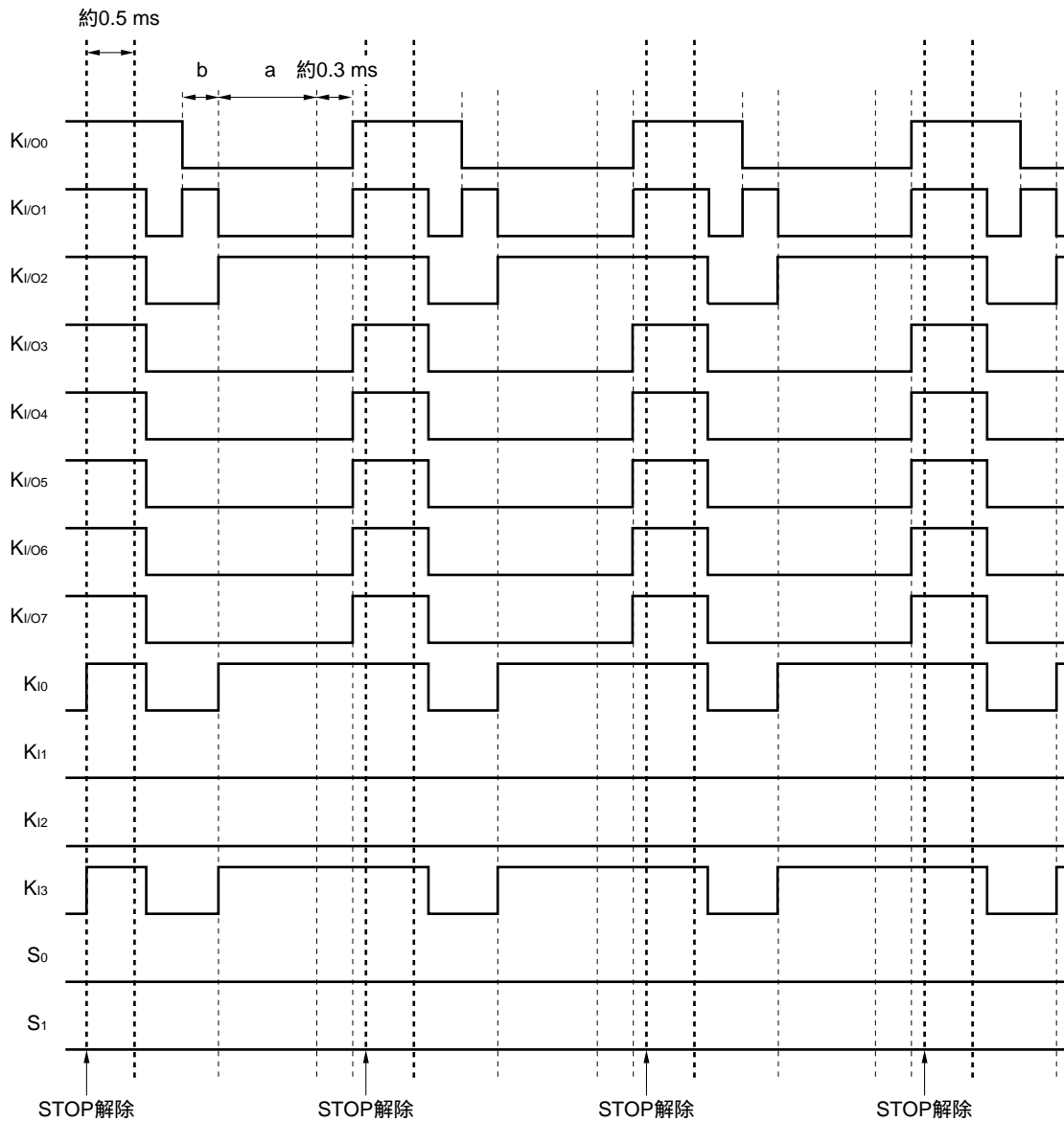
実行時間は約0.41 msです。

c : キー・スキャンを全出力する時間です。この時間は a の時間により変化します。

実行時間は約4.18 ms ~ 4.88 msかかります。

図3 - 3 キー確定までの出力パターン (5/6)

(5) K9とK12が押され、同一キー・ソース間で無効となった場合



a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。

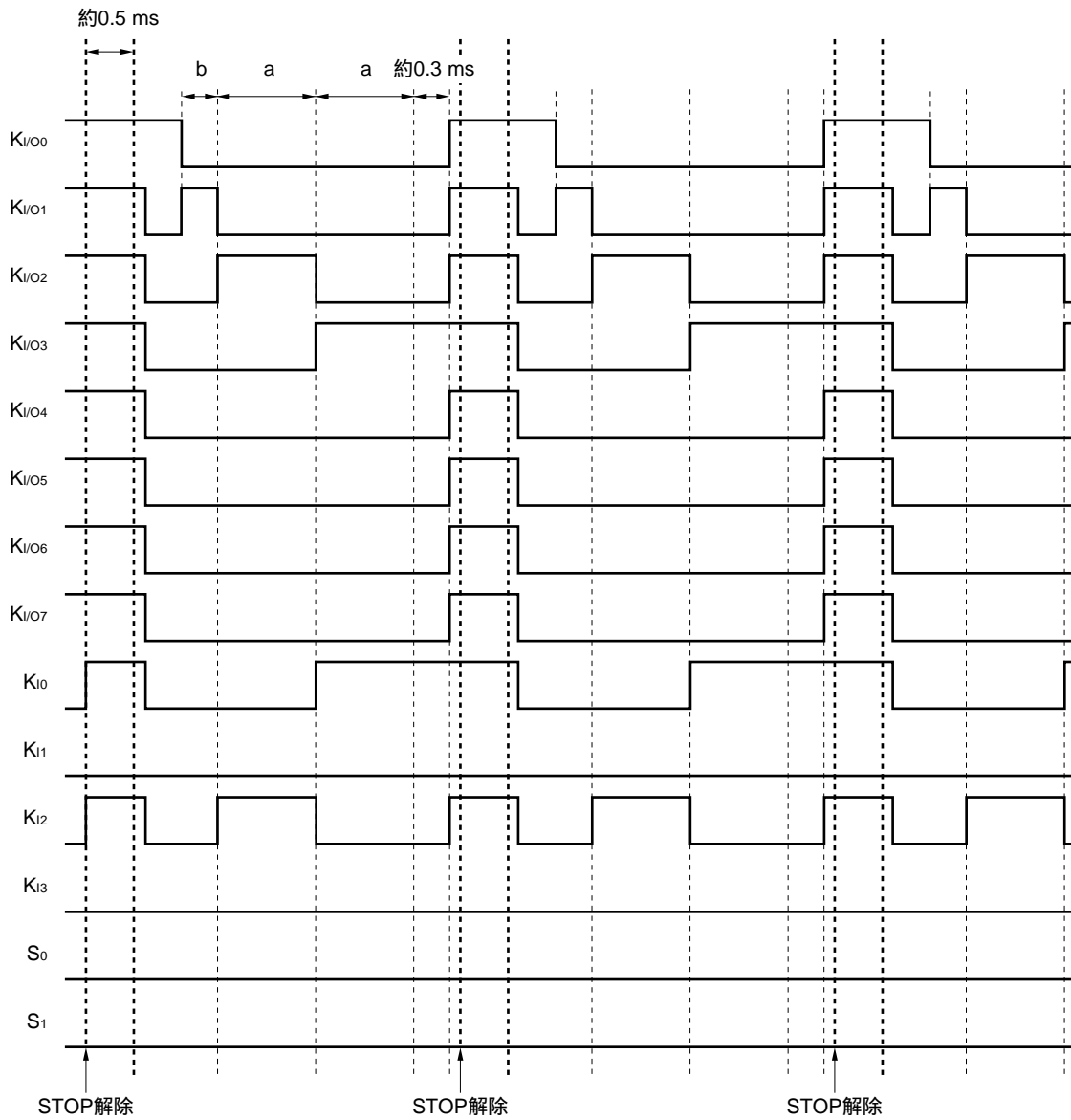
実行時間は約1.32 ms ~ 2.02 msです。

b : キー入力のチェックを行います。

実行時間は約0.41 msです。

図3 - 3 キー確定までの出力パターン (6/6)

(6) K10とK16が押され、異なったキー・ソース間で無効となった場合

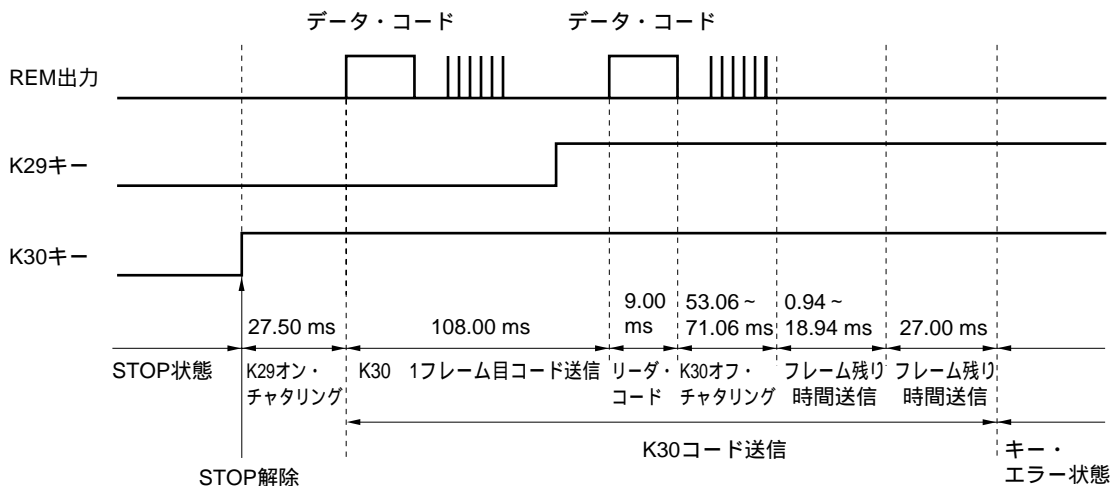


a : キーがオンであると判断した場合、キー・リターンをチェックし、キー・データを算出します。
 実行時間は約1.32 ms ~ 2.02 msです。

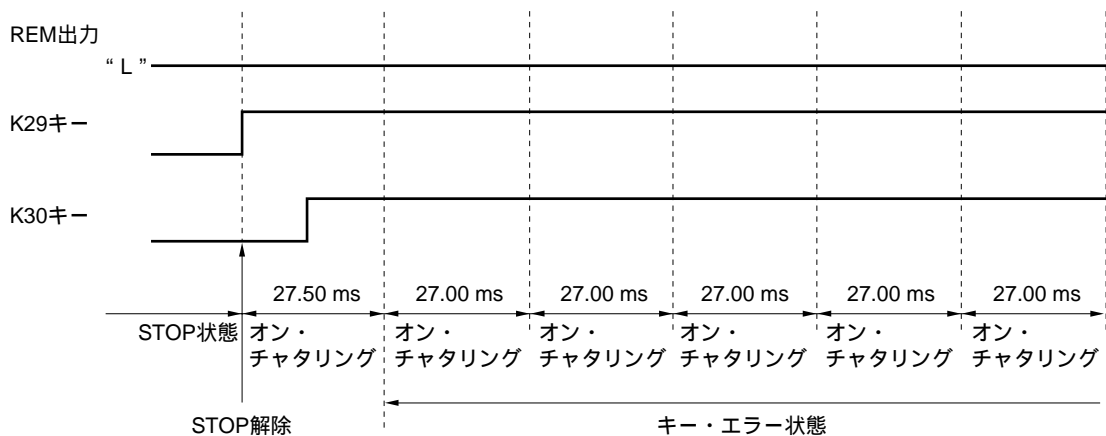
b : キー入力のチェックを行います。
 実行時間は約0.41 msです。

図3 - 4 二重押しキー動作のタイミング・チャート (2/2)

(3) 二重押しの可能なキー (K30-K32) が確定したあとにK29を押した場合、キー・チェックでキー・エラーとなり、コード送信は行われません (二重押しは無効)。



(4) K29が確定する前に二重押しの可能なキー (K30-K32) を押した場合、キー・チェックでキー・エラーとなり、コード送信は行われません (二重押しは無効)。



第4章 出力コード

出力コードを表4 - 1に示します。

NECフォーマットで出力する場合、当社にてカスタム・コードをお客様ごとに割り当てることによって、NECフォーマットを出力する別のリモコンからの干渉を防いでいます。このプログラムでは、カスタム・コードを0AH、カスタム・コード'をF5Hとしています。

カスタム・コードの取得申請については、当社販売員までご相談ください。

表4 - 1 出力コード(1/2)

キーNo.	カスタム・コード	カスタム・コード'	データ	キーNo.	カスタム・コード	カスタム・コード'	データ	キーNo.	カスタム・コード	カスタム・コード'	データ
K1	0AH	F5H	00H	K28	0AH	F5H	1BH	K55	0AH	F5H	36H
K2	0AH	F5H	01H	K29	0AH	F5H	1CH	K56	0AH	F5H	37H
K3	0AH	F5H	02H	K30	0AH	F5H	1DH	K57	0AH	F5H	38H
K4	0AH	F5H	03H	K31	0AH	F5H	1EH	K58	0AH	F5H	39H
K5	0AH	F5H	04H	K32	0AH	F5H	1FH	K59	0AH	F5H	3AH
K6	0AH	F5H	05H	K33	0AH	F5H	20H	K60	0AH	F5H	3BH
K7	0AH	F5H	06H	K34	0AH	F5H	21H	K61	0AH	F5H	3CH
K8	0AH	F5H	07H	K35	0AH	F5H	22H	K62	0AH	F5H	3DH
K9	0AH	F5H	08H	K36	0AH	F5H	23H	K63	0AH	F5H	3EH
K10	0AH	F5H	09H	K37	0AH	F5H	24H	K64	0AH	F5H	3FH
K11	0AH	F5H	0AH	K38	0AH	F5H	25H	K65	0AH	F5H	40H
K12	0AH	F5H	0BH	K39	0AH	F5H	26H	K66	0AH	F5H	41H
K13	0AH	F5H	0CH	K40	0AH	F5H	27H	K67	0AH	F5H	42H
K14	0AH	F5H	0DH	K41	0AH	F5H	28H	K68	0AH	F5H	43H
K15	0AH	F5H	0EH	K42	0AH	F5H	29H	K69	0AH	F5H	44H
K16	0AH	F5H	0FH	K43	0AH	F5H	2AH	K70	0AH	F5H	45H
K17	0AH	F5H	10H	K44	0AH	F5H	2BH	K71	0AH	F5H	46H
K18	0AH	F5H	11H	K45	0AH	F5H	2CH	K72	0AH	F5H	47H
K19	0AH	F5H	12H	K46	0AH	F5H	2DH	K73	0AH	F5H	48H
K20	0AH	F5H	13H	K47	0AH	F5H	2EH	K74	0AH	F5H	49H
K21	0AH	F5H	14H	K48	0AH	F5H	2FH	K75	0AH	F5H	4AH
K22	0AH	F5H	15H	K49	0AH	F5H	30H	K76	0AH	F5H	4BH
K23	0AH	F5H	16H	K50	0AH	F5H	31H	K77	0AH	F5H	4CH
K24	0AH	F5H	17H	K51	0AH	F5H	32H	K78	0AH	F5H	4DH
K25	0AH	F5H	18H	K52	0AH	F5H	33H	K79	0AH	F5H	4EH
K26	0AH	F5H	19H	K53	0AH	F5H	34H	K80	0AH	F5H	4FH
K27	0AH	F5H	1AH	K54	0AH	F5H	35H				

表4 - 1 出力コード(2/2)

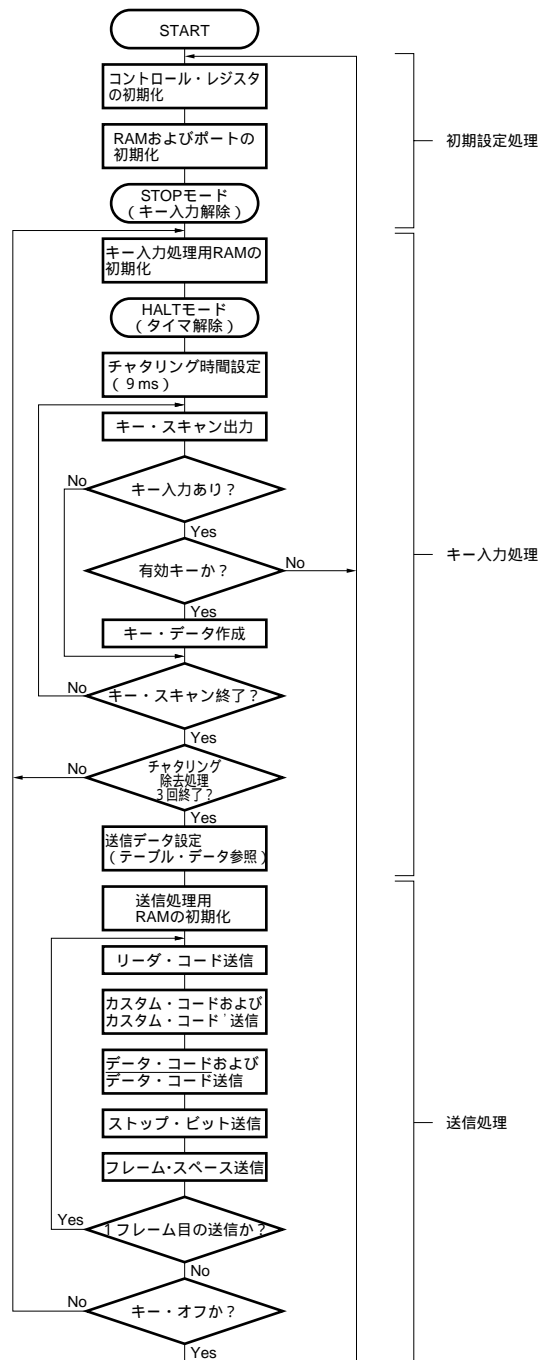
キーNo.	カスタム・コード	カスタム・コード'	データ
K29 + K30	0AH	F5H	90H
K29 + K31	0AH	F5H	91H
K29 + K32	0AH	F5H	92H

第5章 ソフトウェア構成

5.1 ジェネラル・フロー・チャート

このプログラムのジェネラル・フロー・チャートを図5-1に示します。

図5-1 ジェネラル・フロー・チャート



5.2 プログラム・メモリ (ROM) 構成

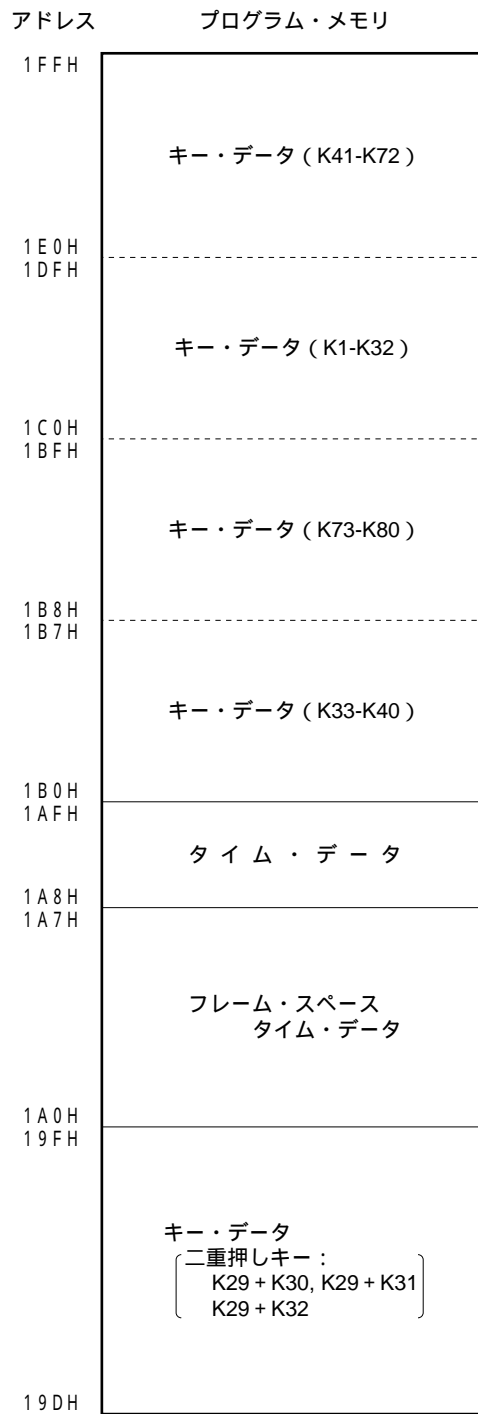
μPD6134のプログラム・メモリ (ROM) は、1002ステップ×10ビット構成になっています (3EAHから3FFH番地の22ステップはテスト・プログラム領域です)。

このプログラムのROMマップを図5 - 2に、データ・テーブル・マップを図5 - 3に示します。

図5 - 2 ROMマップ

アドレス	プログラム・メモリ
3FFH	テスト・プログラム領域
3EAH 3E9H	
20CH 20BH	空 き 領 域
200H 1FFH	フレーム・スペース出力 サブ - チ ン
19DH 19CH 19BH	データ・テーブル (図5 - 3 参照)
0F8H 0F7H	空 き 領 域
013H 012H	送 信 処 理
000H	キ ー 入 力 処 理
	初 期 設 定

図5 - 3 データ・テーブル・マップ



5.3 データ・メモリ (RAM) 構成

データ・メモリ (RAM) は32×4ビット構成のスタティックRAMで、処理データのストアに利用します。命令によっては8ビット単位で操作することもできます。

R0はROMアドレスのデータ・ポインタとしての機能を持っています。このデータ・ポインタにROMアドレスを設定し、ROMの内容を読み出すことができます。これをROMデータのテーブル参照と呼びます。リセット時には、R0は00Hになります。

また、RFはアドレス・スタック・レジスタと兼用になっています。リセット時にはR1～RFは不定となります。

プログラム全体 (キー入力処理 + 送信処理)、キー入力処理、送信処理のRAMマップを表5 - 1から表5 - 3に示します。また、RAM使用説明を表5 - 4に示します。

表5 - 1 RAMマップ

	H (1)	L (0)
R0	ワーク1 データ・ポインタH	チャタリング・カウンタ データ・ポインタL
R1	確定キー・データH	確定キー・データL
R2	イミーディエト (0FH) 設定用	イミーディエト (1H) 設定用
R3	K29 ONフラグ	連続フラグ
R4	キー・データH カスタム・コードH	キー・データL カスタム・コードL
R5	カスタム・コード' H	カスタム・コード' L
R6	キーありフラグ データ・コードH	キー・スキャン・カウンタ データ・コードL
R7	イミーディエト (0FH) 設定用	キー・オフ・チェック・カウンタ
R8	キー・スキャン・データH イミーディエト (0FH) 設定用 データ1	キー・スキャン・データL イミーディエト (0H) 設定用 データ0
R9	比較キー・データH データ1送信カウンタH	比較キー・データL データ1送信カウンタL
RA	ワーク2	キー・リターン・チェック・カウンタ 送信ビット・カウンタ
RB	イミーディエト (0CH) 設定用	イミーディエト (3H) 設定用
RC	イミーディエト (08H) 設定用	イミーディエト (3H) 設定用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	

表5 - 2 キー入力処理時使用RAMマップ

	H (1)	L (0)
R0	ワーク1 データ・ポインタH	チャタリング・カウンタ データ・ポインタL
R1	確定キー・データH	確定キー・データL
R2	イミーディエト (0FH) 設定用	イミーディエト (1H) 設定用
R3	K29 ONフラグ	連続フラグ
R4	キー・データH カスタム・コードH	キー・データL カスタム・コードL
R5	カスタム・コード ' H	カスタム・コード ' L
R6	キーありフラグ データ・コードH	キー・スキャン・カウンタ データ・コードL
R7		
R8	キー・スキャン・データH イミーディエト (0FH) 設定用	キー・スキャン・データL イミーディエト (0H) 設定用
R9	比較キー・データH	比較キー・データL
RA		キー・リターン・チェック・カウンタ
RB	イミーディエト (0CH) 設定用	イミーディエト (3H) 設定用
RC	イミーディエト (08H) 設定用	イミーディエト (2H) 設定用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	

: 送信処理で使用

表5 - 3 送信処理時使用RAMマップ

	H (1)	L (0)
R0	データ・ポインタH ワーク1	データ・ポインタL
R1		
R2		
R3		連続フラグ
R4	カスタム・コードH	カスタム・コードL
R5	カスタム・コード' H	カスタム・コード' L
R6	データ・コードH	データ・コードL
R7	イミューディエト (OFH) 設定用	キー・オフ・チェック・カウンタ
R8	データ1	データ0
R9	データ1送信カウンタH	データ1送信カウンタL
RA	ワーク2	送信ビット・カウンタ
RB		
RC	未使用	未使用
RD	未使用	未使用
RE	未使用	未使用
RF	アドレス・スタック・レジスタ	

: キー入力処理で使用

表5 - 4 RAM使用説明(1/3)

名 称	RAM	説 明						
データ・ポインタ	R0	ROMアドレスを指定するためのポインタです。 ROMデータのテーブル参照を行う際に使用します。						
チャタリング・カウンタ	R00	キー取り込み時のチャタリング回数をカウントするためのカウンタです。 3回カウントするためイニシャル値として、0CHが設定されます。0FHで3回終了と判断します。						
ワーク1	R10	ワーク・エリアです。 データを一時的に格納するためのエリアです。						
確定キー・データ	R1	オン・チャタリング除去処理終了後に、確定したキー・データを格納します。 すでに確定キー・データが格納されている場合には、チャタリング除去処理終了時のキー・データ(R4)と比較して一致すれば、このデータは保持されます。不一致の場合には、新たにキー・データ(R4)の値が格納されません。						
イミューディエト(1H)設定用	R02	上位3ビット・クリア用に使用します。						
イミューディエト(0FH)設定用	R12	0FH設定用に使用します。						
連続フラグ	R03	送信処理で、現在1フレーム目を送信中か、2フレーム目以降を送信中かを判断するためのフラグです。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>2フレーム目以降を送信中</td> </tr> <tr> <td>0H</td> <td>1フレーム目を送信中</td> </tr> </tbody> </table> 0FH：セット，0H：クリア	値	説 明	0FH	2フレーム目以降を送信中	0H	1フレーム目を送信中
値	説 明							
0FH	2フレーム目以降を送信中							
0H	1フレーム目を送信中							
K29 ONフラグ	R13	K29キー(二重押しが可能なキー)が押されたかどうかを判断するためのフラグです。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>K29キーが押された場合</td> </tr> <tr> <td>0H</td> <td>K29以外のキーが押された場合</td> </tr> </tbody> </table> 0FH：セット，0H：クリア	値	説 明	0FH	K29キーが押された場合	0H	K29以外のキーが押された場合
値	説 明							
0FH	K29キーが押された場合							
0H	K29以外のキーが押された場合							
キー・データ	R4	オン・チャタリング中での、前回のキー・データを格納します。 比較キー・データ(R9)の値と比較して一致すれば、このデータは保持されます。不一致の場合または初回には、新たに比較キー・データ(R9)の値が格納されます。						
カスタム・コード	R4	カスタム・コードを格納します。						
カスタム・コード'	R5	カスタム・コード'を格納します。						
データ・コード	R6	データ・コードを格納します。						
キー・スキャン・カウンタ	R06	キー・スキャン回数をカウントするためのカウンタです。 8回カウントする必要があるため、イニシャル値として8H(8回)が設定されます。						

表5 - 4 RAM使用説明 (2/3)

名 称	RAM	説 明										
キーありフラグ	R16	キーが押されているかどうかを判断するためのフラグです。										
		<table border="1"> <thead> <tr> <th>値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>キーが押されている場合</td> </tr> <tr> <td>0H</td> <td>キーが離されている場合</td> </tr> </tbody> </table>	値	説 明	0FH	キーが押されている場合	0H	キーが離されている場合				
		値	説 明									
		0FH	キーが押されている場合									
0H	キーが離されている場合											
0FH：セット，0H：クリア												
キー・オフ・チェック・カウンタ	R07	<p>ビット・データ送信中のキー・オフの状態の回数をカウントするためのカウンタです。</p> <p>10回カウントする必要があるため、イニシャル値として5Hが設定され、0FH（10回連続キー・オフ状態）になった時点でキー・オフ状態が確定します。キー・オフではないと判断された場合は、その時点でカウンタがイニシャル化されます。</p>										
イミディエト（0FH）設定用	R17	各フラグをセットするために使用します。										
キー・スキャン・データ	R8	キー・スキャン出力データを格納します。										
データ0 データ1	R08 R18	<p>ビット・データ‘0’および‘1’を送信するための時間データ（テーブル・データ）の口ウ・アドレスを格納します。</p> <p>データ0：ビット・データ‘0’ データ1：ビット・データ‘1’</p>										
イミディエト（0H）設定用	R08	各フラグをセットするために使用します。										
イミディエト（0FH）設定用	R18	<p>データ反転用に使用します。</p> <p>設定値は0FHです。</p>										
比較キー・データ	R9	オン・チャタリング中のキー・データを格納します。										
データ1送信カウンタ	R9	送信したビット・データ‘1’の数をカウントするためのカウンタです。										
キー・リターン・チェック・カウンタ	R0A	<p>キー・リターン・データにより、キー・データを算出するためのカウンタです。</p> <p>K_i入力の場合は4回、S_i入力の場合は1回カウントする必要があるため、イニシャル値は0CH（4回）または0FH（1回）を設定します。</p>										
送信ビット・カウンタ	R0A	ビット送信処理を行う際、ワーク2（R1A）に格納されたビット・データを、必要なビット数送信するためのカウンタです。1～4回のカウンタです。										
		<table border="1"> <thead> <tr> <th>設定値</th> <th>説 明</th> </tr> </thead> <tbody> <tr> <td>0FH</td> <td>1ビット送信（リーダ・コード，ストップ・ビット送信時）</td> </tr> <tr> <td>0EH</td> <td>2ビット送信（未使用）</td> </tr> <tr> <td>0DH</td> <td>3ビット送信（未使用）</td> </tr> <tr> <td>0CH</td> <td>4ビット送信（カスタム・コード，カスタム・コード’，データ・コード，$\overline{\text{データ・コード}}$送信時）</td> </tr> </tbody> </table>	設定値	説 明	0FH	1ビット送信（リーダ・コード，ストップ・ビット送信時）	0EH	2ビット送信（未使用）	0DH	3ビット送信（未使用）	0CH	4ビット送信（カスタム・コード，カスタム・コード’，データ・コード， $\overline{\text{データ・コード}}$ 送信時）
		設定値	説 明									
		0FH	1ビット送信（リーダ・コード，ストップ・ビット送信時）									
		0EH	2ビット送信（未使用）									
0DH	3ビット送信（未使用）											
0CH	4ビット送信（カスタム・コード，カスタム・コード’，データ・コード， $\overline{\text{データ・コード}}$ 送信時）											

表5 - 4 RAM使用説明(3/3)

名 称	RAM	説 明
ワーク2	R1A	ビット送信を行うための出力データを格納します。
イミューディエト(3H)設定用	R0B	データ判断, 3H設定用に使用します。
イミューディエト(0CH)設定用	R1B	データ判断, 0CH設定, 上位2ビット・セット, 下位2ビット・クリア用に使用します。
イミューディエト(2H)設定用	R0C	ビット1の判断に使用します。
イミューディエト(08H)設定用	R1C	下位3ビット・クリアに使用します。

5.4 フラグ・マップ

各処理中でのフラグの動きを、表5 - 5に示します。

表5 - 5 フラグ・マップ

(1) 連続フラグ (R03)

処 理		R03
初期設定処理		クリア (0H)
キー入力処理	オン・チャタリング除去処理終了時のキー・データが、前に行ったオン・チャタリング除去処理時の確定キー・データと異なる場合	クリア (0H)
送信処理	毎フレームのフレーム・スペース送信中	判断
	1フレーム目のみ、フレーム・スペース送信中	セット (0FH)

判断：その処理中に判断を行っている

(2) K29 ONフラグ (R13)

処 理		R13
初期設定処理		クリア (0H)
キー入力処理	キーありフラグがセットされている場合	判断
	チャタリングが3回終了し、確定キー・データがK29であった場合	セット (0FH)
	チャタリングが3回終了し、確定キー・データがK29、有効二重押しのどちらでもない場合	クリア (0H)
送信処理		-

判断：その処理中に判断を行っている

- : 使用していない

(3) キーありフラグ (R16)

処 理		R16
初期設定処理		-
キー入力処理	RAMの初期化時	クリア (0H)
	キー・リターン・チェックの冒頭	判断
	キー・データ算出後	判断
	キー・リターン・チェック中、キー入力があった場合	セット (0FH)
	キー・スキャン (8回分) 終了後	判断
送信処理		×

判断：その処理中に判断を行っている

- : 使用していない × : 別の用途で使用している

第6章 プログラム説明

6.1 初期値設定

赤外線リモート・コントロール送信機用マイクロコントローラは、通常、電源として電池を使用しています。また赤外線LEDは、発光時に大きな電流を消費します。そのため、赤外線LEDの発光時に電源電圧が大きく変動することがあり、これによってRAMやポートの内容が突然変化する場合があることを想定しておく必要があります。

RAMやポートの突然の変化による誤動作を防ぐため、送信を一回行うごとに初期値を設定し直すようにプログラムを設計します。

6.1.1 処理説明

(1) ポートの設定およびコントロール・レジスタの初期値を設定します。

K_{I/O}ポート (P0)

キー・スキャン出力用の8ビットの入出力ポートです。

初期設定値はFFHです。すべてのビット (K_{I/O0}-K_{I/O7}) はハイ・レベル出力に設定されます。

コントロール・レジスタ0 (P3)

コントロール・レジスタ0の内容を表6-1と表6-2に示します。

初期設定値は13Hです。表中の網掛けのように設定されます。

表6-1 コントロール・レジスタ0 (P3)

ビット		b7	b6	b5	b4	b3	b2	b1	b0
名称		-	-	DP(データ・ポインタ)		TCTL	CARY	MOD ₁	MOD ₀
				DP ₉	DP ₈				
設定値	0	"0"固定	"0"固定	0	0	1/1	ON	表6-2参照	
	1			1	1	1/2	OFF		
リセット時		0	0	0	0	0	0	1	1

b₀, b₁..... REM出力のキャリア周波数とDuty比を指定します。

b₂..... b₀, b₁で指定される周波数のキャリアの有無を指定します。

"0" = ON (キャリアあり), "1" = OFF (キャリアなし, ハイ・レベル)。

b₃..... キャリア周波数とタイマ・クロックの分周比を変更します。

"0" = 1/1 (キャリア周波数: b₀, b₁の指定値, タイマ・クロック: f_x/8)。

"1" = 1/2 (キャリア周波数: b₀, b₁の指定値の1/2, タイマ・クロック: f_x/16)。

表6-2 タイマ・クロックとキャリア周波数の設定

b ₃	b ₂	b ₁	b ₀	タイマ・クロック	キャリア周波数 (Duty比)
0	0	0	0	f _x /8	f _x (Duty 1/2)
		0	1		f _x /8 (Duty 1/2)
		1	0		f _x /12 (Duty 1/2)
		1	1		f _x /12 (Duty 1/3)
	1	x	x	キャリアなし (ハイ・レベル)	
1	0	0	0	f _x /16	f _x /2 (Duty 1/2)
		0	1		f _x /16 (Duty 1/2)
		1	0		f _x /24 (Duty 1/2)
		1	1		f _x /24 (Duty 1/3)
	1	x	x	キャリアなし (ハイ・レベル)	

b₄, b₅.....ROMのデータ・ポインタの上位2ビット (DP₈, DP₉) の指定を行います。

備考1 . x : don't care

2 . : 初期設定値 (13H)

3 . f_x : システム・クロック周波数

コントロール・レジスタ1 (P4)

コントロール・レジスタ1の内容を表6-3に示します。

初期設定値は32Hです。表中の網掛けのように設定されます。

表6-3 コントロール・レジスタ1 (P4)

ビット		b7	b6	b5	b4	b3	b2	b1	b0
名称		-	-	Ki pull-down	S0/S1 pull-down	-	S1/ $\overline{\text{LED}}$ mode	K _{I/O} mode	S0 mode
設定値	0	“ 0 ”固定	“ 0 ”固定	OFF	OFF	“ 0 ”固定	S1	IN	OFF
	1			ON	ON		$\overline{\text{LED}}$	OUT	IN
リセット時		0	0	1	0	0	1	1	0

b0.....S0ポートの入力モードを指定します。“ 0 ” = OFFモード (ハイ・インピーダンス) , “ 1 ” = IN (入力モード)。

b1.....K_{I/O}ポートの入出力モードを指定します。“ 0 ” = IN (入力モード) , “ 1 ” = OUT (出力モード)。

b2.....S1/ $\overline{\text{LED}}$ ポートの入出力モードを指定します。“ 0 ” = S1 (入力モード) , “ 1 ” = $\overline{\text{LED}}$ (出力モード)。

b4.....S0/S1ポート入力モード時のプルダウン抵抗の有無を指定します。“ 0 ” = OFF (なし) , “ 1 ” = ON (あり)。

b5.....Kiポート入力モード時のプルダウン抵抗の有無を指定します。“ 0 ” = OFF (なし) , “ 1 ” = ON (あり)。

備考1 . 出力モード, OFFモード時は, プルダウン抵抗はすべて自動的に切れます。

2 . : 初期設定値 (32H)

(2) RAMの初期値を設定します。

次に示すRAMが0にクリアされます。

- ・ 確定キー・データ (R1)
- ・ 連続フラグ (R03)
- ・ K29 ONフラグ (R13)

(3) STOPモードを設定します。

HALT命令の解除条件一覧表を表6-4に示します。

初期設定値は8Hです。初期設定時(STOPモード)の解除条件は、表中の網掛けのように設定されます。

表6-4 HALT命令の解除条件一覧表

HALT命令のオペランド値				設定モード	設定のための前提条件	解除条件
b ₃	b ₂	b ₁	b ₀			
0	0	0	0	STOP	K _{I/O} 端子がすべてハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
	0	1	1	STOP	K _{I/O} 端子がすべてハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
	1	1	0	STOP ^{注1}	K _{I/O0} 端子がハイ・レベル出力	K _I 端子の少なくとも1つにハイ・レベルを入力
1	上記のb ₂ b ₁ b ₀ の組み合わせのうちいずれか			STOP	【上記のほかに次の条件が追加される】 ----- -	S ₀ , S ₁ 端子の少なくとも1つにハイ・レベルを入力 ^{注2}
0/1	1	0	1	HALT	-	タイマのダウン・カウンタが0になったとき

注1 . HALT #x110Bを設定する場合には、暴走時に内部リセットがかかるようにK_{I/O0}端子とK_I端子を使ってキー・マトリクスを構成してください。

2 . S₀, S₁端子の少なくともどちらか1端子(スタンバイ解除に使用する端子)が、入力モードでなければなりません(ただし、両方とも出力モードの場合であっても内部リセットはかかりません)。

注意1 . 上記以外のオペランド値でHALT命令が実行された場合や、HALT命令の実行時に前提条件が満足されていない場合には、内部リセットがかかります。

2 . タイマのダウン・カウンタが0でないとき(タイマ動作中)にSTOPモードを設定すると、タイマのダウン・カウンタとタイマ出力の許可フラグの全10ビットが0にクリアされてから、STOPモードになります。

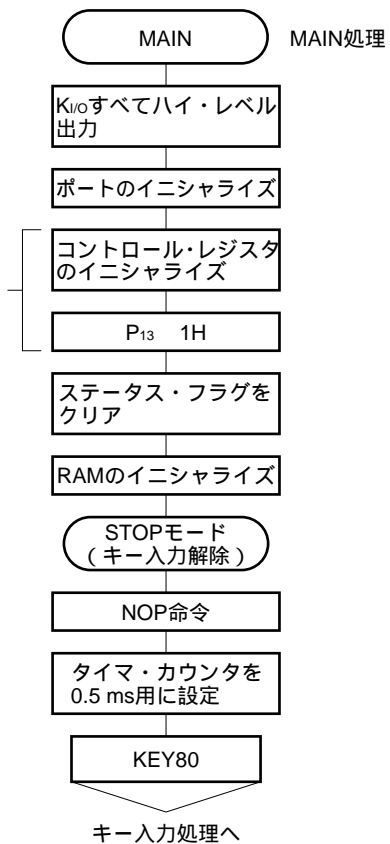
3 . STOPモード解除後の最初の命令はNOP命令を記述してください。

備考 : 初期設定値(8H)

(4) STOPモード解除後、NOP命令を実行してから、タイマに初期値を設定します。

タイマ初期設定値は1CH(=0.5ms)です。

6.1.2 ディテール・フロー・チャート




```

☆☆☆☆☆☆☆☆ 初期設定処理 ☆☆☆☆☆☆☆☆
TIME05M EQU    01CH          ; 0.5ms(0.510ms)
DamyTime EQU    512-1        ; 9.00ms

;##### P U B L I C #####
PUBLIC MAIN
;
;##### S T A R T #####
;*****
; Control Register Initialize
;*****
MAIN:
    OUT    P0,#0FFH          ① キー・スキャン出力 (K1/0) をすべてハイ・レベルに設定します。
    OUT    P4,#032H          ② ポートのインチャライズを行います。
                                S0:ハイ・インピーダンス, S1:入力モード, K1/0:出力モード
    OUT    P3,#013H          ③ コントロール・レジスタ0, 1 (P3, P4) のインチャライズをします。
                                キャリアあり, 周波数: fx/12, デューティ: 1/3,
                                TCTL: 1/1, Ki:プルダウン抵抗あり,
                                S0, S1:プルダウン抵抗あり,
                                データ・ポインタ: DP3, DP4=01H (テーブル参照用アドレスの最上位桁)

    MOV    T,#DamyTime      }----- ④ ステータス・フラグをクリアします。
    STTS   #0101B

;*****
; RAM Initialize Routine
;*****
    MOV    R1,#000H          ⑤ RAMのインチャライズを行います。
                                確定キー・データ (R1) : 00H
    MOV    R3,#000H          K29 ON フラグ (R13), 連続フラグ (R03) : 0H

    HALT   #008H            ⑥ STOPモード: KiおよびS0, S1ハイ・レベル入力により解除されます。
    NOP                                ⑦ NOP命令
    MOV    T,#TIME05M        ⑧ タイマ値のインチャライズを行います。
    ;                                ICH = 0.5ms
END

```

6.2 キー入力処理

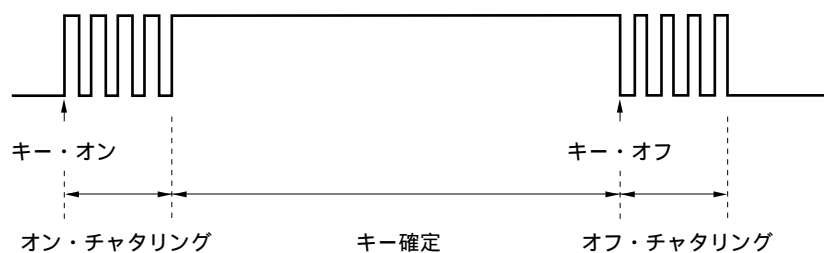
6.2.1 処理説明

キー入力処理はチャタリング除去処理，キー・スキャン処理，カスタム・コード作成処理，キー・データ作成処理により構成されています。

(1) チャタリング除去処理

キーをオンまたはオフにする場合，キー信号がハイ・レベルまたはロウ・レベルに安定するまでの間に，チャタリング（オン・チャタリング，オフ・チャタリング）と呼ばれる不安定な状態が存在します（図6-1を参照）。キー入力は，このチャタリング中では不安定な状態であるため，プログラムでチャタリングを除去する必要があります。

図6-1 キー入力信号のチャタリング



次にチャタリング除去の例（キー・オン判定の例）を図6-2に示します。

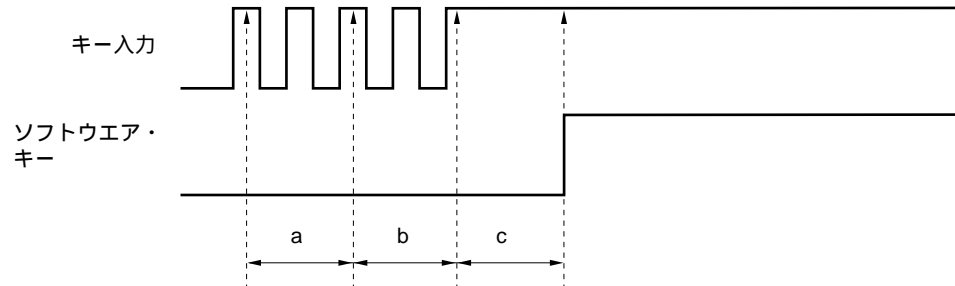
でキーがオンされたことをSTOPモード解除によって認識し，その後一定間隔（ \sim : 9.00 ms \times 3回）で，キー入力をチェックします。

(a)では，チェック位置ですべてキー・オンされているので，ソフトウェア・キーの の位置でキー信号がハイ・レベルになったと判断します。

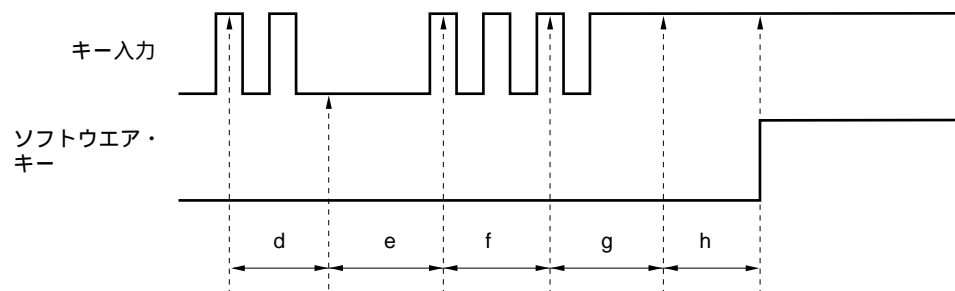
でキーがオンされたと認識してから，ソフトウェア上でハイ・レベルになったと判断するまでの待ち時間（ $a + b + c$ ）をチャタリング除去時間といいます。（a）では，キー入力をチェックするとき常にオンされていたので，チャタリング除去時間は $a + b + c$ になります。（b）では，チャタリング除去時間は $d + e + f + g + h$ になります。

図6-2 チャタリング発生時のキー・オン判定

(a)



(b)



(2) キー・スキャン処理

キー・スキャン処理を , に示します。

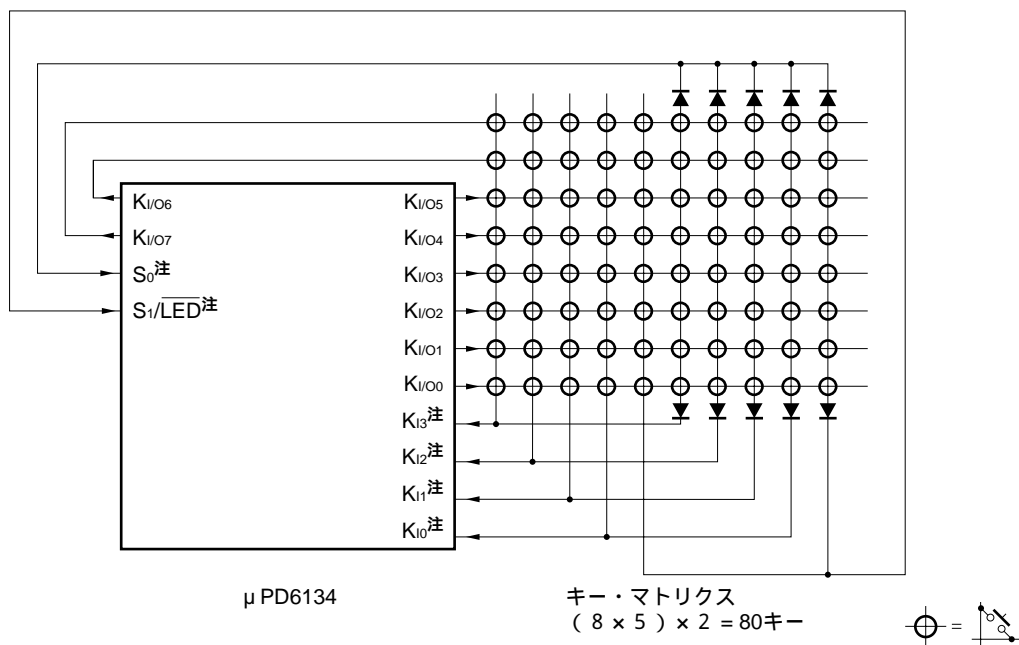
キー・マトリクス

図6 - 3 に示す80個のキー・マトリクスを例にとって説明します。

この例では, $K_{I/O0}$ - $K_{I/O7}$ は出力ポートとしてキー・スキャン信号を出力します。この信号を K_{I0} - K_{I3} , S_0 , S_1 より入力ポートとして取り込みます(キー・リターン信号)。

キー・リターン信号入力ポート(K_{I0} - K_{I3} , S_0 , S_1)には, 内蔵プルダウン抵抗がそれぞれ接続されているので, キーが押されていないときにはロウ・レベルが入力されます。

図6 - 3 80キー・マトリクス



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

キー・スキャン

80キー・マトリクスのうち, どのキーが押されたか判断するには, まず, STTS命令で K_{I0} - K_{I3} , および S_0 , S_1 に入力(ハイ・レベル)があったかを確認します。

次に, キー・スキャン信号出力用ポート($K_{I/O0}$ - $K_{I/O7}$)を $K_{I/O0}$ から順に1つずつハイ・レベルにしていき, $K_{I/O0}$ - $K_{I/O7}$ のどの出力に対して入力があったかを確認します。

プログラム中のキー・スキャン処理部分では, キー・スキャン・カウンタ(キー・ソース位置), およびキー・リターン・チェック・カウンタ(キー・リターン位置)により, キー・データ(キー位置)が判断されます。キー・データと2つのカウンタ値の対応を図6 - 5 に示します。

データ・メモリ上でのキー・データのビット構成については、(4) キー・データのビット構成に示します。

キー・スキャンを行う場合、キーボード上の浮遊容量、配線インピーダンスによりキー・ソースの立ち上がりが遅くなることを考慮する必要があります。このプログラムでは、キー・ソース出力をハイ・レベルで出力してから、100 μs程度(455 kHzの場合6ステップ)待ったあとに、キー入力を取り込むようにしています。

図6-4 キー・マトリクス

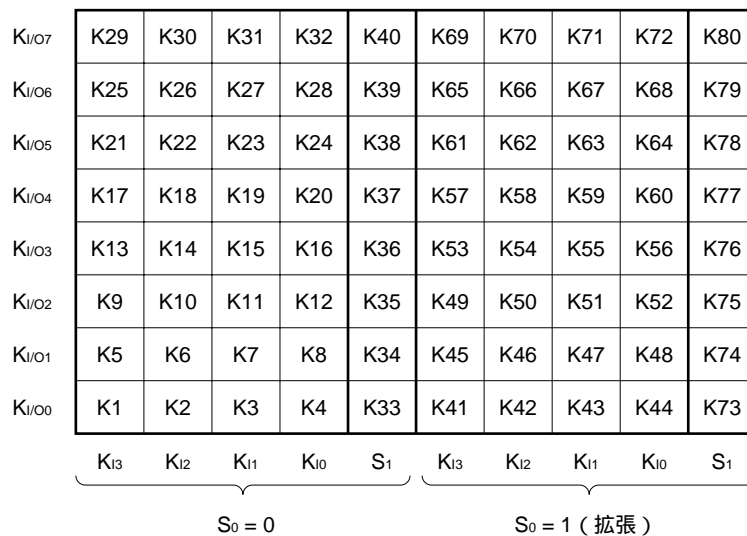


図6-5 キー・データ(キー位置)

キー・スキャン・カウンタ(R06)

F	K29	K30	K31	K32	K40	K69	K70	K71	K72	K80
E	K25	K26	K27	K28	K39	K65	K66	K67	K68	K79
D	K21	K22	K23	K24	K38	K61	K62	K63	K64	K78
C	K17	K18	K19	K20	K37	K57	K58	K59	K60	K77
B	K13	K14	K15	K16	K36	K53	K54	K55	K56	K76
A	K9	K10	K11	K12	K35	K49	K50	K51	K52	K75
9	K5	K6	K7	K8	K34	K45	K46	K47	K48	K74
8	K1	K2	K3	K4	K33	K41	K42	K43	K44	K73

キー・リターン・
チェック・カウンタ(R0A)

C D E F F C D E F F

(3) カスタム・コード作成処理

このプログラムでは、カスタム・コードとして0AH、カスタム・コード'としてF5Hを送信するように設定しています。

具体的にはカスタム・コードとしてデータ・メモリR4に50H、カスタム・コード'としてR5にAFHを設定しています。

注意 NECフォーマットでは、LSBから送信するため、ビット列の並びを逆にした値を設定します。

(4) キー・データ作成処理

キー・データ処理を、 に示します。

キー・データのビット構成

キー・データは8ビットで構成されています。各ビットはキー・ソース、キー・リターンの状態を表します。

キー・スキャン・カウンタはK1/00-K1/07の8種類のため、K10-K13の入力の場合にはキー・データHの最下位ビットとキー・データLの上位2ビットを、S1の入力の場合にはキー・データLの下位3ビットを使用します。

キー・リターン・チェック・カウンタは、K10-K13の入力の場合には4種類のため、キー・データの下位2ビットを使用します。S1の入力の場合には1種類しかいないため、使用しません。

また、キー拡張用のS0のデータは、K10-K13に入力があった場合には、キー・データHのビット1を、S1に入力があった場合にはキー・データLの最上位ビットを使用します。

最初にキー入力があるかを判断したときに、K1入力以外（S1入力）の場合にはキー・データHに0BHを設定します。このキー・データHは、図6-6、図6-7のどちらのフォーマットで作成するかを判断するときにも使用します。

また、このキー・データは、 **データ・コード**に示すように、データ・コード算出するときの、テーブル参照用アドレスとして使用します。そのため、図6-6、図6-7に示したようなビット構成でキー・データを作成します。

図6-6 K10-K13に入力があったときのキー・データのビット構成

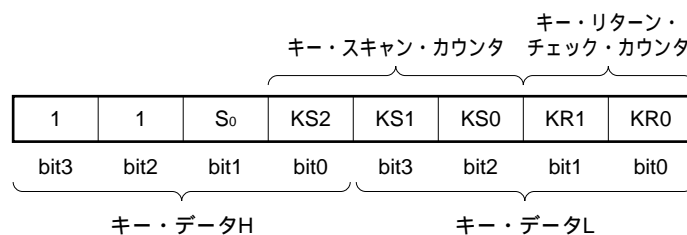
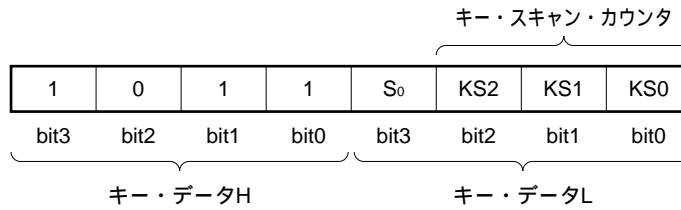


図6-7 S1に入力があつたときのキー・データのビット構成



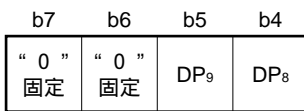
データ・コード

図6-6, 図6-7に示している, キー入力処理によって得られたキー・データを, テーブル参照アドレスとして使用することにより, データ・コードを算出することができます。

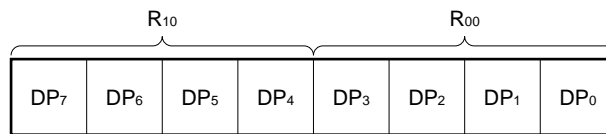
テーブル参照方法は, コントロール・レジスタ0 (P3) の上位4ビットに1Hを設定し, キー・データをデータ・ポインタ (図6-8参照) に設定することにより, ROM内容を送信コードとして使用することができます。

図6-8 データ・ポインタの構成

コントロール・レジスタ0 (P3)

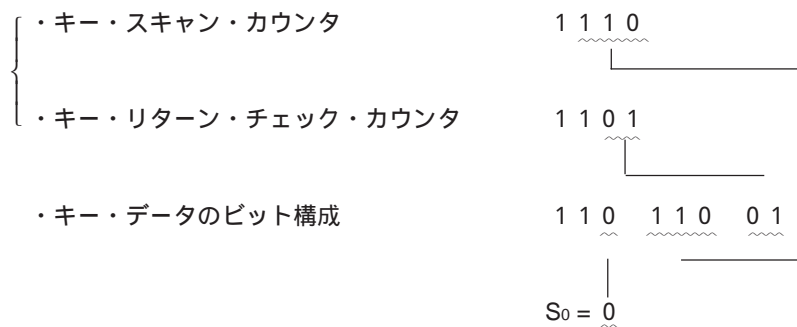


データ・ポインタ (DP)



例 キー位置: K26 (キー入力: K12) の場合

1. キー・スキャン・カウンタ (R06) = EH,
キー・リターン・チェック・カウンタ (R0A) = 0Hになります (図6-5参照)。
2. キー・リターン・カウンタの下位3ビットとキー・リターン・チェック・カウンタの下位2ビットを使用し, 図6-6に示したようなビット構成でキー・データを作成します。



3. キー・データをデータ・ポインタに設定します。

・データ・ポインタの構成 0 0 0 1 1 1 0 1 1 0 0 1 1 D 9 H

図6-6, 図6-7のビット構成でキーを作成すると, 次のようなテーブル参照アドレスになります (なお, 前述のとおり, コントロール・レジスタ0 (P3) の上位4ビットには1Hが設定されています)。

・キー入力: K_{10} - K_{13} , キー位置: $K1$ - $K32$ ($S_0 = 0$ の場合)

...参照アドレス = 1C0H-1DFH

・キー入力: K_{10} - K_{13} , キー位置: $K41$ - $K72$ ($S_0 = 1$ の場合)

...参照アドレス = 1E0H-1FFH

・キー入力: S_1 , キー位置: $K33$ - $K40$ ($S_0 = 0$ の場合)

...参照アドレス = 1B0H-1B7H

・キー入力: S_1 , キー位置: $K73$ - $K80$ ($S_0 = 1$ の場合)

...参照アドレス = 1B8H-1BFH

・二重押し ($K29 + K30$, $K29 + K31$, $K29 + K32$)

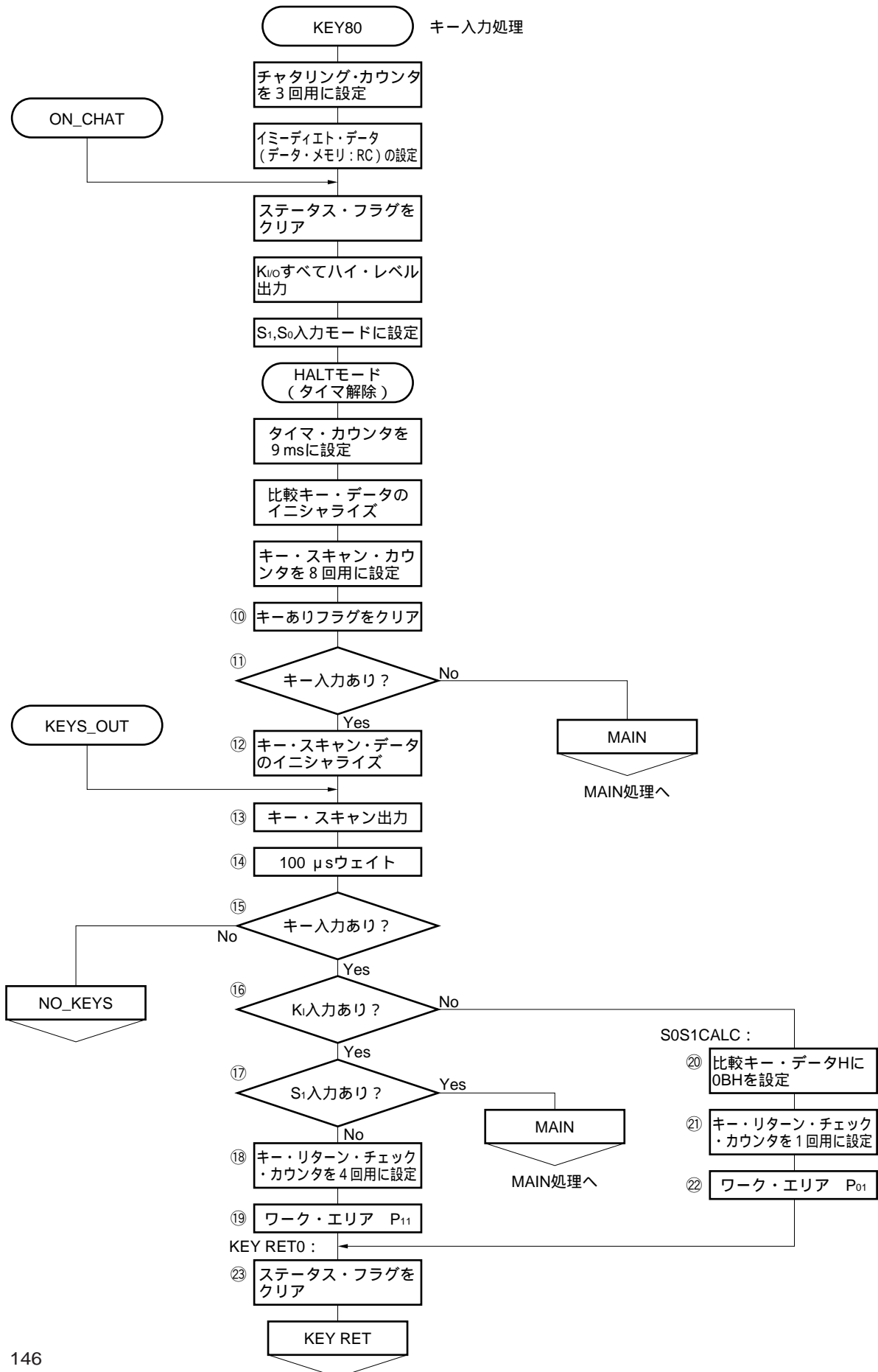
二重押しが確定した場合, キー・データHは9Hを設定します。

$K29 + K30$...参照アドレス = 19DH

$K29 + K31$...参照アドレス = 19EH

$K29 + K32$...参照アドレス = 19FH

6.2.2 ディテール・フロー・チャート



☆☆☆☆☆☆ キー入力処理 ☆☆☆☆☆☆

```

TIME9M EQU 1FFH ; 9.00ms(9.002ms)
CUSTM1 EQU 050H ; Custom Code = 0AH
CUSTM2 EQU 0AFH ; Custom Code' = F5H

```

```

;##### P U B L I C #####
PUBLIC KEY80
;

```

```

;##### E X T E R N #####
EXTRN MAIN ; MAIN Routine
;

```

```

;##### S T A R T #####
KEY80:

```

```
MOV R0, #00CH
```

① チャタリング・カウンタ (R0) を設定 (=0CH) にします
(0FHで3回カウント終了)。

```
MOV RC, #082H
```

② RCをイミディエト・データ設定用に使用 (R1c=8H, R0c=2H) します。

```

;*****
; ON Chattering
;*****

```

```
ON_CHAT:
```

```
STTS #0101B
```

③ ステータス・フラグをクリアします。

```
OUT P0, #0FFH
```

④ キー・スキャン出力 (K_{1/o}) をすべてハイ・レベルに設定します。

```
OUT P4, #033H
```

⑤ S₁, S₀を入力モードに設定します。

```
HALT #005H
```

⑥ HALTモード：タイマ・カウンタが00Hになると解除されます。

```
MOV T, #TIME9M
```

⑦ タイマ・カウンタを設定します。9ms=1FFH

```
MOV R9, #000H
```

⑧ 比較キー・データ (R9) のイニシャライズ (00H) を行います。

```
MOV R6, #008H
```

⑨⑩ キー・スキャン・カウンタ (R6) を8H, キーありフラグ (R1e) を0Hに設定します。

```
STTS #1110B
```

⑪ キー入力 (K₁, S₀, S₁) があるか判断します。

```
JNF MAIN
```

キー入力なしの場合, MAINへ

```
MOV R8, #001H
```

⑫ キー・スキャン・データ (R8) のイニシャライズ (01H) を行います。

```

;*****
; Key Scan
;*****

```

```
KEYS_OUT:
```

```
MOV A, R08
```

```
OUT P00, A
```

```
MOV A, R18
```

```
OUT P10, A
```

⑬ キー・スキャン・データ (P0) を出力します。

```

;+++++
; ++ キー入力まで100μs Wait ++
;+++++

```

```
MOV R2, #0F1H
```

[備考] R2をイミディエト・データ設定用に使用 (R12:0FH, R02:1H) します。

```
MOV RB, #0C3H
```

[備考] RBをイミディエト・データ設定用に使用 (R1B:0CH, R0B:3H) します。

```
NOP
```

⑭ キー入力チェックを行うまで100μs待ちます。

```
NOP
```

```
NOP
```

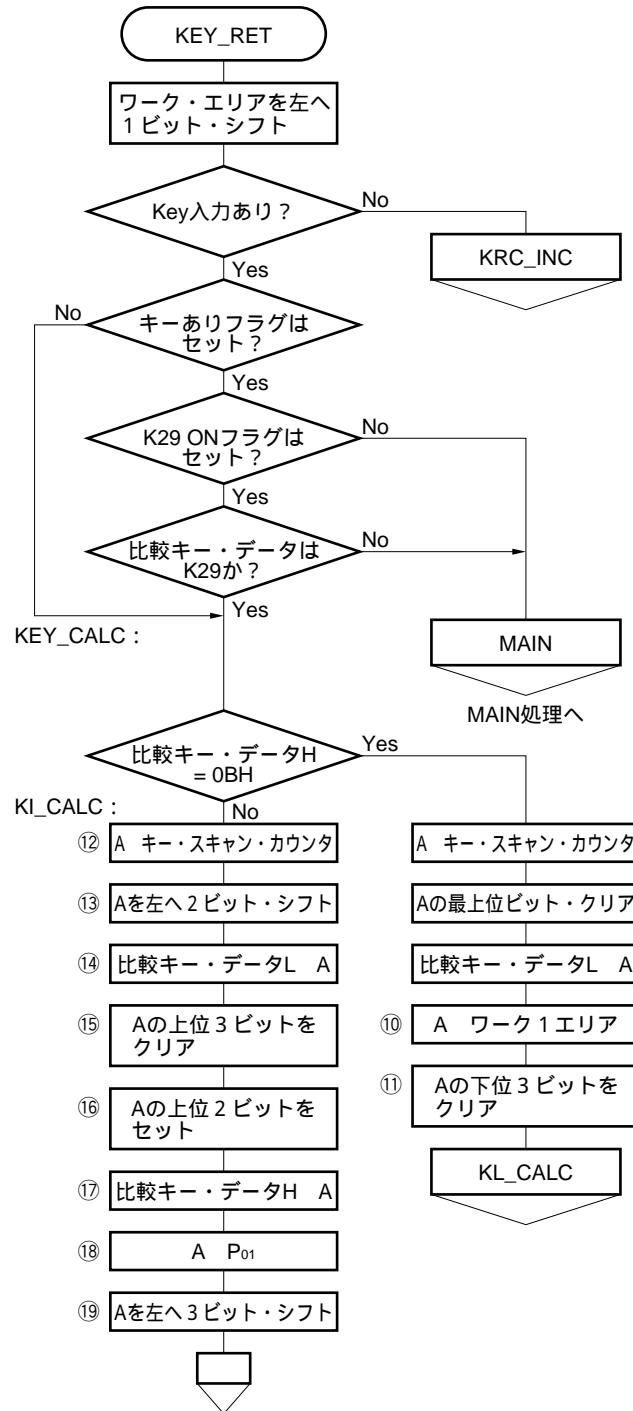
```
NOP
```

```

;*****
;           Key Return Check
;*****
      STTS  #1011B           ⑮ キー入力 (Ki, S0, S1) があるかどうかを判断します。
      JNF   NO_KEYS         キー入力なしの場合, NO_KEYSへ
      STTS  #0011B           ⑯ Ki入力かS0, S1入力かを判断します。
      JNF   SOS1CALC        S0, S1の場合, SOS1CALCへ
;*****
;           KI Data Calculate
;*****
      IN    A, P01           ⑰ Ki入力があり, S1入力もあるかどうか判断します。
      RL    A                S1入力がある場合, 多重押しエラーと判断し, MAINへ
      JC    MAIN
      MOV   A, R1B           ⑱ キー・リターン・チェック・カウンタ (R0A) のイニシャライズを行います。
      MOV   ROA, A           0CH=4回 (このときR1B=0CH)
      IN    A, P11           ⑲ アキュムレータにKiの端子状態 (P11) を転送します。
      JMP   KEY_RETO
;*****
;           S0,S1 Data Calculate
;*****
SOS1CALC:
      MOV   R9, #0B0H        ⑳ 比較キー・データH (R19) にK33~K40, K73~K80 (S0, S1入力) 判断用の値
                               (0BH) を設定します。

      MOV   A, R12
      MOV   ROA, A           ㉑ キー・リターン・チェック・カウンタ (R0A) のイニシャライズを行います。
                               0FH=1回用 (このときR12=0FH)
      IN    A, P01           ㉒ アキュムレータにS0, S1の端子状態 (P01) を転送します。
KEY_RETO:
      MOV   R10, A           [備考] ワーク・エリアにアキュムレータの値 (P00またはP01) を格納します。
      STTS  #0101B           ㉓ ステータス・フラグをクリアします。

```



```

KEY_RET:
MOV   A, R10
RLZ   A
MOV   R10, A
JNC   KRC_INC
; ** Key入力あり **
MOV   A, R16
SCAF  A
JNC   KEY_CALC
MOV   A, R13
SCAF  A
JNC   MAIN
MOV   A, R0C
XRL   A, R19
SCAF  A
JNC   MAIN
MOV   A, R09
XRL   A, R0B
SCAF  A
JNC   MAIN
; *****
;   Key Data Calculate
; *****
KEY_CALC:
MOV   A, #04H
XRL   A, R19
SCAF  A
JNC   KI_CALC
; ** S1データ算出 **
MOV   A, R06
ANL   A, #0111B
MOV   R09, A
MOV   A, R10
ANL   A, R1C
JMP   KL_CALC
;
; ** KIデータ算出 **
KI_CALC:
MOV   A, R06
RL    A
RL    A
MOV   R09, A
ANL   A, R02
ORL   A, R1B
MOV   R19, A
IN    A, P01
RL    A
RL    A
RL    A

```

① ワーク1 (R10) を左へ1ビット・シフトします。

② キー入力がある (CY=1) かを判断します。
キー入力がない場合、KRC_INCへ

③ キーが押されている (二重押し) かを判断します。
押されていない (単押し) 場合、KEY_CALCへ

④ K29 ON フラグがセットされているかを判断します。
クリアされている場合、MAINへ
[備考] このとき、R0c=2H

⑤ 比較キー・データ (R9) はK29のキー・データと一致するかを判断します。
一致していない場合、MAINへ

⑥ 押されたキーがK33~K40, K73~K80 (S0, S1入力) かどうかを判断します。
K1~K32, K41~K72のキーが押された場合、アキュムレータの値 K1_CALCへ

⑦ キー・スキャン・カウンタ (R0e) をアキュムレータに転送します。

KS3	KS2	KS1	KS0
-----	-----	-----	-----

⑧ アキュムレータの最上位ビットをクリアします。

0	KS2	KS1	KS0
---	-----	-----	-----

⑨ 比較キー・データL (R0a) にアキュムレータの内容を格納します。
Acc→R0a

⑩ ワーク1エリア (R10) をアキュムレータに転送します。

S0	1	1	S1
----	---	---	----

⑪ アキュムレータの下位3ビットをクリアします (このときR1c=8H)。
KL_CALCへ

S0	0	0	0
----	---	---	---

⑫ キー・スキャン・カウンタ (R0e) をアキュムレータに転送します。

KS3	KS2	KS1	KS0
-----	-----	-----	-----

⑬ アキュムレータを左へ2ビット・シフトします。

KS1	KS0	KS3	KS2
-----	-----	-----	-----

⑭ アキュムレータの値を比較キー・データL (R0a) に格納します。
Acc→R0a

⑮ アキュムレータの上位3ビットをクリアします (このときR02=1H)。

1	0	0	KS2
---	---	---	-----

⑯ アキュムレータの上位2ビットをセットします (このときR1b=0CH)。
Acc→R1a

1	1	0	KS2
---	---	---	-----

⑰ 比較キー・データH (R1a) にアキュムレータの内容を格納します。

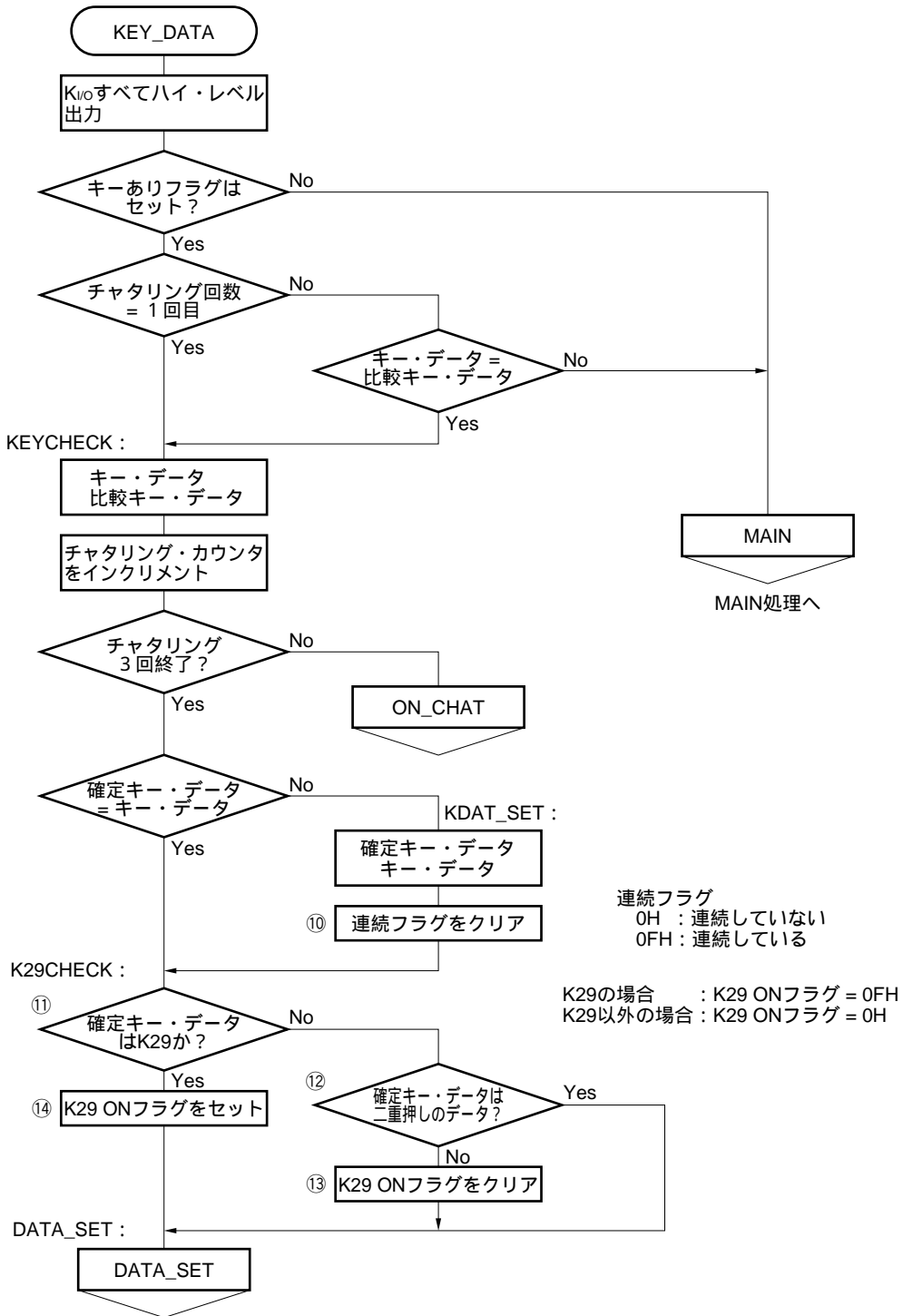
⑱ アキュムレータにS1, S0のデータを転送します。

S1	S0	1	1
----	----	---	---

⑲ アキュムレータを左へ3ビット・シフトします。

1	S1	S0	1
---	----	----	---

		アキュムレータの値						
ANL	A, R0C	①	アキュムレータのビット3, ビット2, ビット0をクリアします (このときR0C=2H)。	<table border="1"><tr><td>0</td><td>0</td><td>S0</td><td>0</td></tr></table>	0	0	S0	0
0	0	S0	0					
ORL	A, R19	②	アキュムレータと比較キー・データH (R19)をORします。	<table border="1"><tr><td>1</td><td>1</td><td>S0</td><td>KS2</td></tr></table>	1	1	S0	KS2
1	1	S0	KS2					
MOV	R19, A	③	比較キー・データH (R19) にアキュムレータの内容を格納します。	Acc→R19				
MOV	A, R09	④	アキュムレータに比較キー・データL (R09)を転送します。	<table border="1"><tr><td>KS1</td><td>KS0</td><td>KS3</td><td>KS2</td></tr></table>	KS1	KS0	KS3	KS2
KS1	KS0	KS3	KS2					
ANL	A, R1B	⑤	アキュムレータの下位2ビットをクリアします (このときR1B=1100B)。	<table border="1"><tr><td>KS1</td><td>KS0</td><td>0</td><td>0</td></tr></table>	KS1	KS0	0	0
KS1	KS0	0	0					
MOV	R09, A	⑥	比較キー・データL (R09) にアキュムレータの内容を格納します。	Acc→R09				
MOV	A, R0A	⑦	アキュムレータにキー・リターン・チェック・カウンタ (R0A)を転送します。	<table border="1"><tr><td>KR3</td><td>KR2</td><td>KR1</td><td>KR0</td></tr></table>	KR3	KR2	KR1	KR0
KR3	KR2	KR1	KR0					
ANL	A, R0B	⑧	アキュムレータの上位2ビットをクリアします (このときR0B=0011B)。	<table border="1"><tr><td>0</td><td>0</td><td>KR1</td><td>KR0</td></tr></table>	0	0	KR1	KR0
0	0	KR1	KR0					
KL_CALC:				K1入力の場合				
ORL	A, R09	⑨	アキュムレータと比較キー・データL (R09)をORします。	<table border="1"><tr><td>KS1</td><td>KS0</td><td>KR1</td><td>KR0</td></tr></table> S1入力の場合	KS1	KS0	KR1	KR0
KS1	KS0	KR1	KR0					
				<table border="1"><tr><td>S0</td><td>KS2</td><td>KS1</td><td>KS0</td></tr></table>	S0	KS2	KS1	KS0
S0	KS2	KS1	KS0					
MOV	R09, A	⑩	比較キー・データL (R09) にアキュムレータの内容を格納します。	Acc→R09				
;** 二重押し Check **								
MOV	A, R16	}	⑪ キーありフラグ (R16) がセットされているかを判断します。クリアの場合, SINGLE_Kへ					
SCAF								
JNC	SINGLE_K							
MOV	A, #09H	}	⑫ 比較キー・データH (R19) に9Hを設定します。					
MOV	R19, A							
SINGLE_K:								
MOV	A, R12	}	⑬ キーありフラグ (R16) をセットします (このときR12=0FH)。					
MOV	R16, A							
KRC_INC:								
MOV	A, R0A	}	⑭ キー・リターン・チェック・カウンタ (R0A) をインクリメントします。					
INC	A							
MOV	R0A, A							
JNC	KEY_RET		⑮ キー・リターン・チェックが, K1入力の場合は4回, S1入力の場合は1回終了したかを判断します。まだ終了していない場合, KEY_RETへ					
NO_KEYS:								
STTS	#0101B		⑯ ステータス・フラグをクリアします。					
MOV	A, R06	}	⑰ キー・スキャン・カウンタ (R06) をインクリメントします。					
INC	A							
MOV	R06, A							
JC	KEY_DATA		⑱ キー・スキャンが8回終了したかを判断します。終了の場合, KEY_DATAへ					
MOV	A, R18	}	⑲ キー・スキャン・データ (R8) を左へ1ビット・シフトします。					
RL	A							
MOV	R18, A							
MOV	A, R08							
RL	A							
MOV	R08, A							
JNC	KEYS_OUT							
MOV	R8, #010H							
JMP	KEYS_OUT							




```

;*****
; Transmit Key Data = ON Chattering Key Data ??
;*****
KEY_DATA:
    OUT    P0,#0FFH          ① キー・スキャン出力 (K10) をすべてハイ・レベルに設定します。
    MOV    R8,#0F0H          [備考] R8をイミューディエト・データ設定用に使用 (R18=0FH, R08=0H) します。

    MOV    A, R16
    SCAF   }
    JNC   MAIN }----- ② キーありフラグ (R16) がセットされているかを判断します。
    MOV    A, R00             クリアの場合, MAINへ
    XRL   A, R0B             }
    SCAF   }----- ③ チャタリング処理が1回目かを判断します (このときR08=#0011B)。
    JC    KEYCHECK           1回目の場合, KEYCHECKへ

    MOV    A, R14
    XRL   A, R18             [備考] このときR18=0FH
    XRL   A, R19
    SCAF   }
    JNC   MAIN }----- ④ キー・データ (R4) と比較キー・データ (R9) が一致しているかを判断しま
    XRL   A, R04             す。
    XRL   A, R09             (このときアキュムレータ=0FH)
    SCAF   }----- 不一致の場合はMAINへ
    JNC   MAIN

KEYCHECK:
    MOV    A, R19
    MOV    R14, A           }----- ⑤ 比較キー・データ (R9) をキー・データ (R4) に格納します。
    MOV    A, R09
    MOV    R04, A
    MOV    A, R00
    INC    A                }----- ⑥ チャタリング・カウンタ (R00) をインクリメントします。
    MOV    R00, A
    SCAF   }
    JNC   ON_CHAT          ⑦ チャタリング処理が3回終了したかを判断します。
    JNC   ON_CHAT          まだ終了していない場合, ON_CHATへ

; ** Key High Check **
    XRL   A, R11
    XRL   A, R14           }----- [備考] このときアキュムレータ=0FH
    SCAF   }
    JNC   KDAT_SET

; ** Key Low Check **
    XRL   A, R01
    XRL   A, R04           }----- ⑧ 確定キー・データ (R1) とキー・データ (R4) は一致しているかを判断しま
    SCAF   }----- 不一致の場合, KDAT_SETへ
    JC    K29CHECK        一致の場合, K29CHECKへ

KDAT_SET:
    MOV    A, R14
    MOV    R11, A          }----- ⑨ キー・データ (R4) を確定キー・データ (R1) に格納します。
    MOV    A, R04
    MOV    R01, A
    MOV    A, R08
    MOV    R03, A          }----- ⑩ 連続フラグ (R03) をクリアします (このときR08=0H)。

```

K29CHECK:		
MOV	A, R0C	[備考] このときR _{0c} =2H ⑪ K29が押されたかを判断します (このときR _{0b} =3H)。 K29以外のキーが押された場合, K29_FLGへ
XRL	A, R11	
SCAF		
JNC	DBL_K_CHK	
MOV	A, R01	
XRL	A, R0B	
SCAF		
JC	K29_FLG	
DBL_K_CHK:		
MOV	A, #06H	⑫ 確定キー・データは二重押しのデータかを判断します。 二重押しのデータの場合, DATA_SETへ
XRL	A, R11	
SCAF		
JC	DATA_SET	
MOV	A, R08	⑬ アク्यूムレータに0Hをセットします (このときR _{0a} =0H)。
K29_FLG:		
MOV	R13, A	⑬⑭ K29 ON フラグをクリアします (このときアク्यूムレータ=0H)。 K29 ON フラグをセットします (このときアク्यूムレータ=0FH)。

DATA_SET

カスタム・コード
テーブル・データ

カスタム・コード
テーブル・データ

データ・ポインタ
確定キー・データ

テーブル参照
データ・コード
テーブル・データ

TRNS

送信処理へ

```

;*****
;          Transmit Data Set
;*****
DATA_SET:
  ** Custom Code **
  MOV     R4,#CUSTM1           ① カスタム・コード (R4) にテーブル・データを格納します。
  ** Custom Code' **
  MOV     R5,#CUSTM2           ② カスタム・コード' (R5) にテーブル・データを格納します。

  ** Data Code **
  MOV     A,R11                }
  MOV     R10,A                }----- ③ 確定キー・データ (R1) をデータ・ポインタ (R0) に設定します。
  MOV     A,R01                }
  MOV     R00,A                }
  MOV     R6,@R0                ④ データ・コード (R6) にテーブル・データを格納します。

SET_END:
  END

```

6.3 送信処理

送信処理は、データ・メモリに格納されたカスタム・コード、カスタム・コード'、データ・コード、データ・コードを第2章 送信波形で示したNEC-Rフォーマットに従い送信します。

送信方法は、送信時間をデータ・テーブルで持ち、テーブル参照によりタイマ・カウンタに設定します。その後、HALTモード（解除条件：TIMER）に入ることにより送信が可能になります。

時間管理を正確および簡単に行うために、図6-10のような方法を行います。HALT解除直後に、次の送信時間データを上記と同様の方法でタイマ・カウンタに設定することにより、HALT-HALT間が設定時間で動作することになります。

キャリア出力あり（以下‘H’とする）を設定するときには、タイマ・カウンタの最上位ビット（出力制御ビット）をセットします。また、キャリア出力なし（以下‘L’とする）を設定するときには、タイマ・カウンタの最上位ビット（出力制御ビット）をクリアします。

タイマ・カウンタの構成を図6-9に示します。

図6-9 タイマ・カウンタの構成

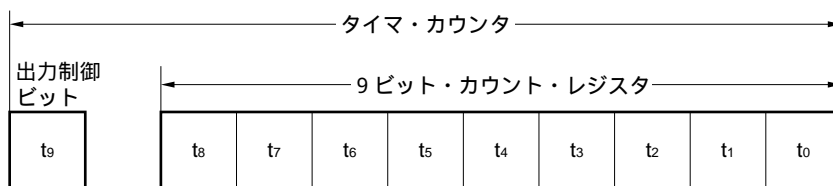
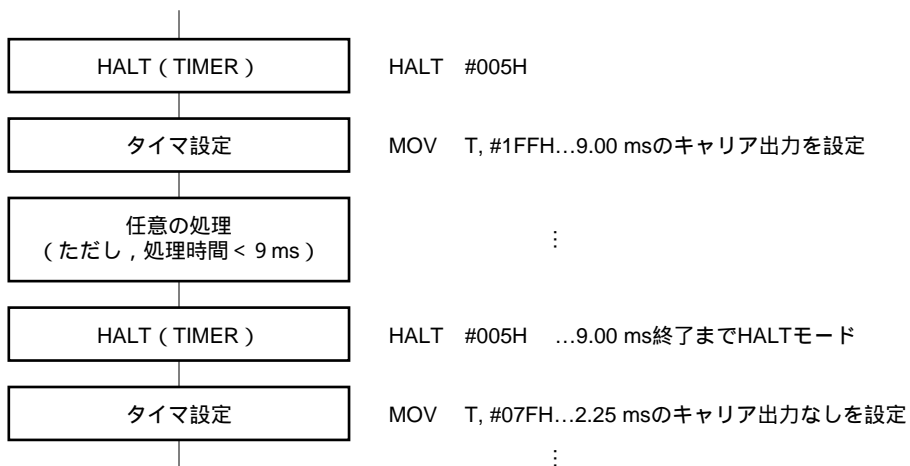


図6-10 タイマ・カウンタの時間管理



タイマの動作時間は「(設定値 + 1) × 8 / fx」で求めることができます。

455 kHzで動作している場合、「MOV T,#1FFH」は「(1FFH + 1) × 8 / (455 × 10³) 9.00 ms」を設定したことになります。

6.3.1 処理説明

送信処理は次に示す7つの処理により構成されています。

- ・初期設定
- ・リーダー・コードの送信
- ・コード送信（カスタム・コード，カスタム・コード'，データ・コード，データ・コード）
- ・ストップ・ビットの送信
- ・フレーム・スペースの送信
- ・2フレーム以降の送信
- ・オフ・チャタリング除去処理

次に，各処理について説明します。

(1) 初期設定

送信処理の初期設定内容を次の ， に示します。

キー・オフ・チェック・カウンタ（R07）を10回（= 05H）に設定します。

データ1送信カウンタを0DFH（24ビット用）に設定します。

カウンタの内訳を表6-5に示します。

このデータ1送信カウンタは，本来フレーム・スペースの送信時間に影響を与えるカスタム・コード，カスタム・コード'内の'データ1'の個数をカウントするためのカウンタです。

データ・コード，データ・コード内の'データ1'の個数は固定（8個）のためフレーム・スペースの送信時間に影響を与えません。

表6-5 データ1送信カウンタ（R9）の内訳

カウンタ値	データ1送信カウンタの内訳
D0H E6H	'データ1'の個数：0～7個 データ・コード，データ・コード内の'データ1'の個数は固定（8個）のため，この状態は存在しない
E7H	'データ1'の個数：8個 カスタム・コード，カスタム・コード'の'データ1'の個数：0個
E8H EFH	'データ1'の個数：9～16個 カスタム・コード，カスタム・コード'の'データ1'の個数：1～8個
F0H F7H	'データ1'の個数：17～24個 カスタム・コード，カスタム・コード'の'データ1'の個数：9～16個

(4) ストップ・ビットの送信

ストップ・ビットの送信方法を次の ~ に示します。

ストップ・ビット用のテーブル参照用アドレス (1AEH) を次のように設定します。

・データ・ポインタH (R10) に上位アドレス (0AH)

・データ0 (R08) にビット・データの 'データ0' の下位アドレス (0EH)

ストップ・ビット用の出力データ0Hをワーク2 (R1A) に設定します。

ストップ・ビットを1ビット送信としてサブルーチンをコールします。

ワーク2 (R1A) を左シフトさせてビット判断を行います。

判断されたビットのテーブル参照用アドレスLを、データ・ポインタLに設定することにより、送信時間をタイマ・カウンタに設定することができます。

ストップ・ビット (H : 0.56 ms , L : 3.00 ms) を送信します。

(5) フレーム・スペースの送信

コード送信時間は、ビット・データの 'データ1' と 'データ0' で送信時間に差があるため、59.06 ms ~ 77.06 msの範囲で変化します。そのためコード送信を行う場合には、フレーム・スペースの送信時間が、'データ1' の個数によって変化します。

ただし、データ・コード、データ・コード内の 'データ1' の個数は固定 (8個) のためフレーム・スペースの送信時間には影響しません。フレーム・スペースの送信時間に影響を与えるのは、カスタム・コード、カスタム・コード内の 'データ1' の個数です。

送信時間の変化に対応するため、送信時に 'データ1' の個数をカウントします。このカウント値によりフレーム・スペースの送信時間を調節します。'データ1' のカウント処理は、(3) コード送信を参照してください。

'データ1' の個数に対するフレーム・スペースの時間を表6-6に示します。

また、タイマ・カウンタは設定可能な最大時間が9.00 msのため、図6-11 ~ の3パターンのように、フレーム・スペースを分割して送信を行います。

表6-6と図6-11との差は、コード送信を行ったときの誤差 ('データ0' のロウ・レベル = 0.565 msを0.56 msに設定) をフレーム・スペースで吸収するためです。

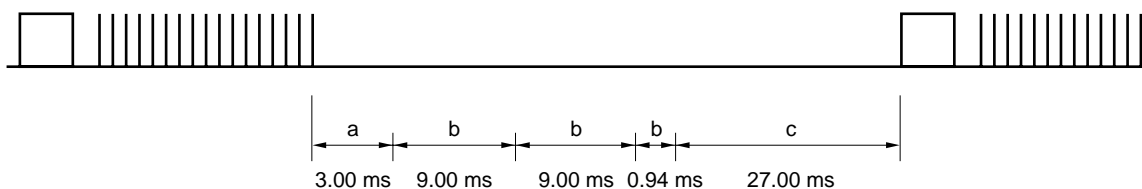
表6-6 ‘データ1’の個数に対するフレーム・スペースの時間

データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)	データ1の 個数(個)	送信時間 (ms)
8	48.940	13	43.315	18	37.690	23	32.065
9	47.815	14	42.190	19	36.565	24	30.940
10	46.690	15	41.065	20	35.440		
11	45.565	16	39.940	21	34.315		
12	44.440	17	38.815	22	33.190		

備考 データ・コード、データ・コード内の‘データ1’の個数は固定(8個)のため、‘データ1’が0~7個の場合はありません。

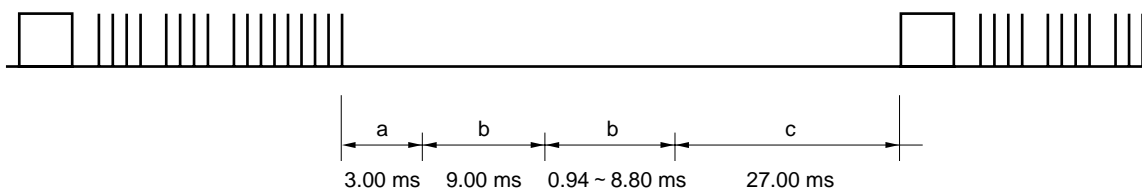
図6-11 フレーム・スペース送信時間の分割方法(1/2)

‘データ1’が8個(データ1送信カウンタ値:0E7H)の場合



- a : ストップ・ビット送信後、フレーム・スペースを3.00 ms送信します。
- b : ‘データ1’の個数に対する時間(18.94 ms)、フレーム・スペースを送信します。
- c : キーを押し続けた場合には、オン・チャタリング除去処理(9.00 ms x 3回)を行います。
それ以外の場合には、フレーム・スペースを27.00 ms送信します。

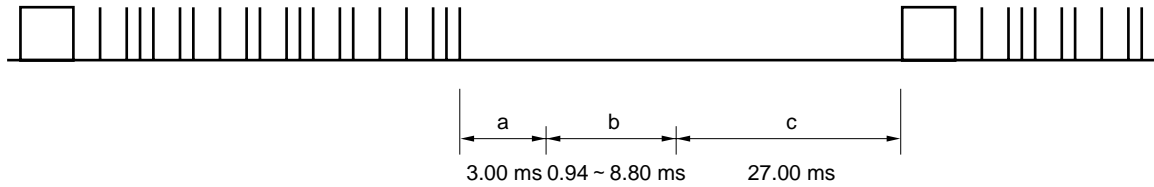
‘データ1’が9~16個(データ1送信カウンタ値:0E8H~0EFH)の場合



- a : ストップ・ビット送信後、フレーム・スペースを3.00 ms送信します。
- b : ‘データ1’の個数に対する時間(9.94 ~ 17.80 ms)、フレーム・スペースを送信します。
- c : キーを押し続けた場合には、オン・チャタリング除去処理(9.00 ms x 3回)を行います。
それ以外の場合には、フレーム・スペースを27.00 ms送信します。

図6-11 フレーム・スペース送信時間の分割方法 (2/2)

‘データ1’が17~24個(データ1送信カウンタ値:0F0H~0F7H)の場合



- a : ストップ・ビット送信後, フレーム・スペースを3.00 ms送信します。
- b : ‘データ1’の個数に対する時間(0.94 ~ 8.80 ms), フレーム・スペースを送信します。
- c : キーを押し続けた場合には, オン・チャタリング除去処理(9.00 ms x 3回)を行います。
それ以外の場合には, フレーム・スペースを27.00 ms送信します。

(6) 2フレーム以降の送信

キーが離されるかキーが変更されるまで, 1フレーム目と同様のコードの送信を, リーダ・コードから繰り返します。送信方法は, (2) ~ (5)を参照してください。

キー・オフのチェックについては, (7) オフ・チャタリング除去処理を参照してください。

(7) オフ・チャタリング除去処理

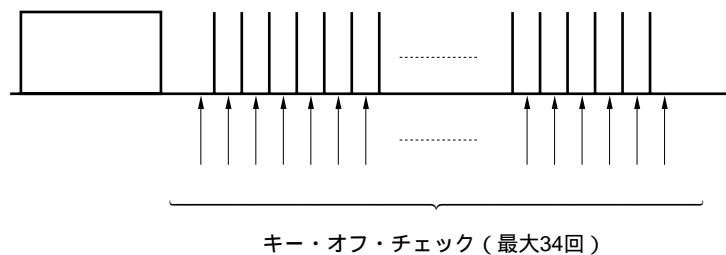
ビット・データのロウ・レベル出力中にキー・オフのチェックを行います。

リーダー・コード, ストップ・ビットを含め34回のロウ・レベル出力中に, 10回連続でキー入力なしと判断した場合のみ, キー・オフが確定します。10回連続チェック中に1回でもキー入力ありと判断した場合には, 10回連続用のカウンタはクリアされ, 最初からカウントを開始します。

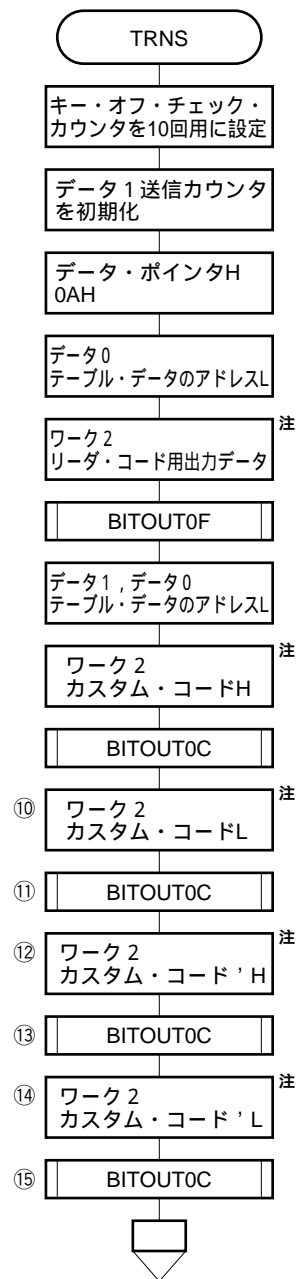
34回のチェックでキー・オフが確定しなかった場合, キーが押し続けられていると判断します。

1フレーム送信中にキー・オフが確定した場合でも, 最低2フレームの送信を行ってから初期設定処理を行います。

図6-12 ビット・データ送信時のキー・オフ・チェック



6.3.2 ディテール・フロー・チャート



注 メイン・ルーチンでは、リーダ・コード用出力データをアキュムレータ（A）に設定し、サブルーチン（BITOUT0C）でワーク2（R1A）にそのデータを設定します。

☆☆☆☆☆☆☆☆ 送信処理 ☆☆☆☆☆☆☆☆

TIME9M EQU 1FFH ; 9.00ms(9.002ms)

P U B L I C #####
;

E X T E R N #####
EXTRN MAIN ; MAIN Routine
EXTRN KEY80 ; Key Check Routine
EXTRN FS_27MS ; Frame Space Subroutine

S T A R T #####
TRNS:

MOV R7, #0F5H ① キー・オフ・チェック・カウンタ (R07) を10回 (=5H) に設定します。
R17をイミューディエト設定用に使用 (R17=0FH) します。

MOV R9, #0DFH ② データ1送信カウンタ (R09) をイニシャライズ (DFH) します。

; Leader Code

[備考] リード・コードの送信 (H:9.00ms, L:4.50ms)

MOV R0, #0A0H ③ データ・ポインタH (R10) に0AH (ビット・データ送信時間用テーブル・
データの上位アドレス) を設定します。

MOV R8, #008H ④ データ1 (R18), データ0 (R08) にビット・データ送信時間用テーブル・
データの低位アドレスを設定します (R18←0H, R08←8H)。

MOV A, R18 ⑤ アキュムレータにリード・コード用の出力データ (0H) を設定します
(アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。

CALL BITOUTOF ⑥ ビット出力サブルーチンをコールします (1ビット送信用: カウンタ値=0FH)。

; Custom Code

[備考] カスタム・コードの送信 (H:0.56ms, L:0.56msまたは1.69ms)

MOV R8, #0CAH ⑦ データ1 (R18), データ0 (R08) にカスタム・コード用のテーブル・データの下
位アドレスを設定します (R18←CH, R08←AH)。

MOV A, R14 ⑧ アキュムレータにカスタム・コード用の出力データH (R14) を設定します
(アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。

CALL BITOUTOC ⑨ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。

MOV A, R04 ⑩ アキュムレータにカスタム・コード用の出力データL (R04) を設定します
(アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。

CALL BITOUTOC ⑪ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。

; Custom Code'

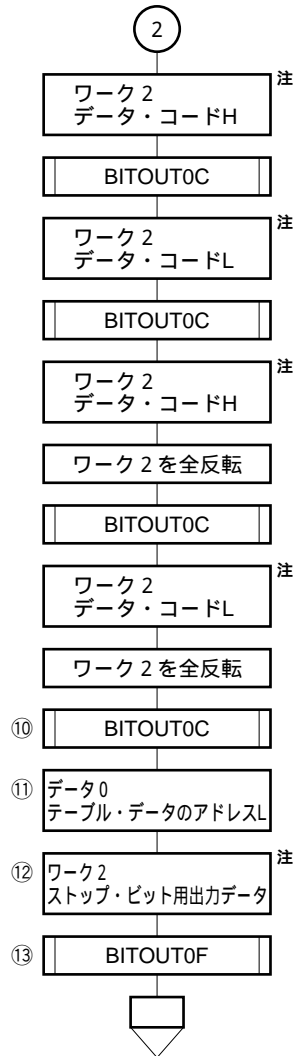
[備考] カスタム・コード' の送信 (H:0.56ms, L:0.56msまたは1.69ms)

MOV A, R15 ⑫ アキュムレータにカスタム・コード' 用の出力データH (R15) を設定します
(アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。

CALL BITOUTOC ⑬ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。

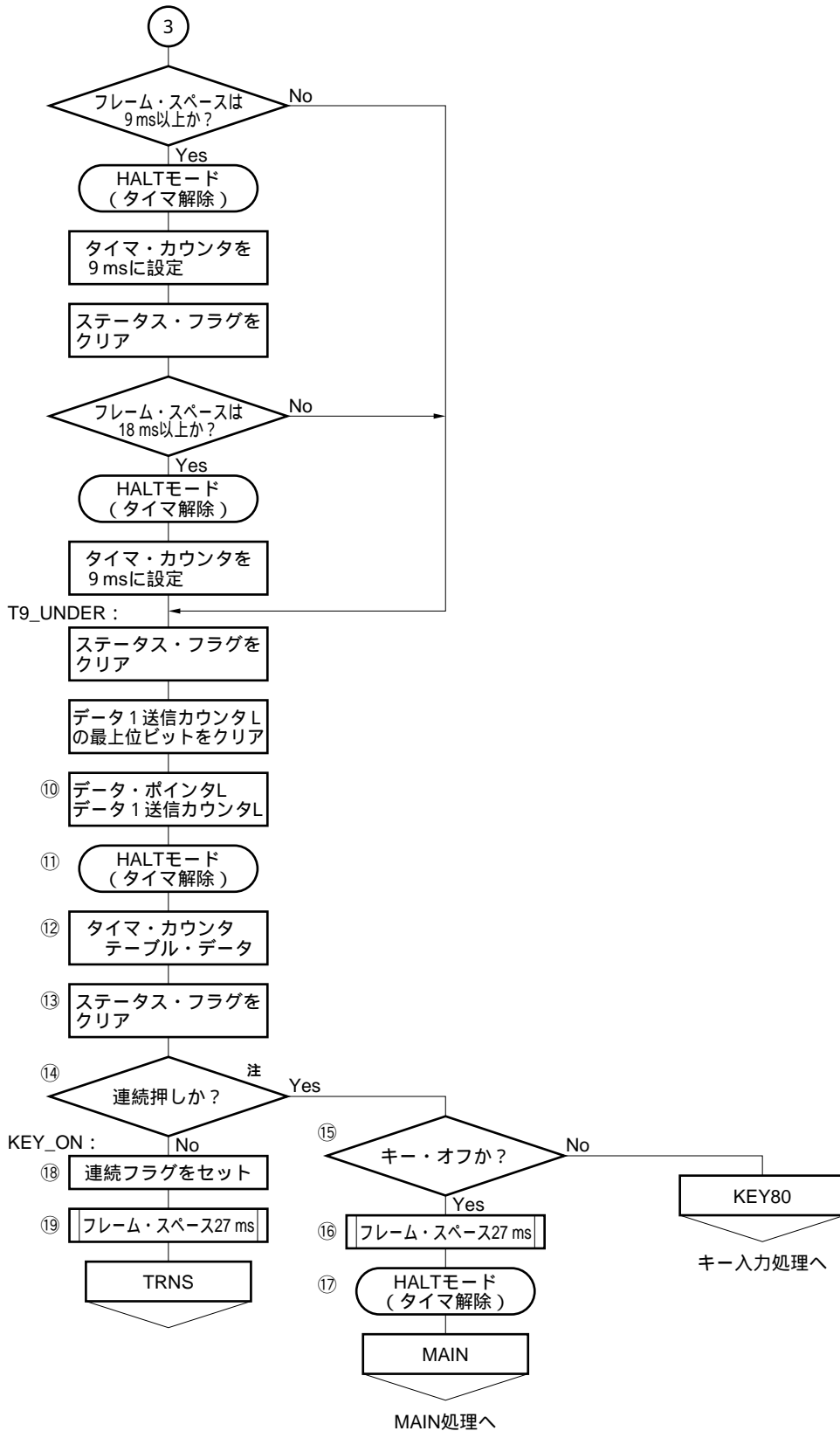
MOV A, R05 ⑭ アキュムレータにカスタム・コード' 用の出力データL (R05) を設定します
(アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。

CALL BITOUTOC ⑮ ビット出力サブルーチンをコールします (4ビット送信用: カウンタ値=0CH)。



注 メイン・ルーチンでは、リーダ・コード出力データをアキュムレータ（A）に設定し、サブルーチン（BITOUT0C）でワーク2（R1A）にそのデータを設定します。

;*****		
:	Data Code	[備考] データ・コードの送信 (H:0.56ms, L:0.56msまたは1.69ms)
;*****		
MOV	A, R16	① アキュムレータにデータ・コード用の出力データH (R16)を設定します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	② ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
MOV	A, R06	③ アキュムレータにデータ・コード用の出力データL (R06)を設定します。 (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	④ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
;*****		
:	Data Code'	[備考] データ・コードの送信 (H:0.56ms, L:0.56msまたは1.69ms)
;*****		
MOV	A, R16	⑤ アキュムレータにデータ・コード用の出力データH (R16)を設定します。
XRL	A, #0FH	⑥ アキュムレータの値を反転させ、データ・コードHの出力データを作成します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	⑦ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
MOV	A, R06	⑧ アキュムレータにデータ・コード用の出力データL (R06)を設定します。
XRL	A, #0FH	⑨ アキュムレータの値を反転させ、データ・コードLの出力データを作成します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0C	⑩ ビット出力サブルーチンをコールします (4ビット送信用:カウンタ値=0CH)。
;*****		
:	Stop Bit	[備考] ストップ・ビットの送信 (H:0.56ms, L:3ms)
;*****		
MOV	R8, #00EH	⑪ データ0 (R18) にストップ・ビットのキャリアありのテーブル・データの低位 アドレスを設定します。
MOV	A, R18	⑫ アキュムレータにストップ・ビット用の出力データ (0H)を設定します (アキュムレータの値はビット出力サブルーチン内でワーク2に転送します)。
CALL	BITOUT0F	⑬ ビット出力サブルーチンをコールします (1ビット送信用:カウンタ値=0FH)。



注 連続押しであれば，連続フラグ = 0FH

```

;*****
;                               [備考] フレーム・スペースの送信 (L:30.94ms~48.94ms)
;*****
MOV    A, R19 }-----① カスタム・コード (16ビット) のビット・データ '1' の数が8個以下か判断
SCAF   }-----      します。
JC     T9_UNDER }-----      9個以上の場合, T9_UNDERへ

HALT   #005H }-----② HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV    T, #TIME9M }-----③ 8個以下の場合, タイマ・カウンタに9ms (=1FFH) を設定します。
STTS   #0101B }-----④ ステータス・フラグをクリアします。

MOV    A, R09 }-----⑤ カスタム・コード (16ビット) のビット・データ '1' の数が, 0個かを判断
RL     A }-----      します。
JC     T9_UNDER }-----      1~8個の場合, T9_UNDERへ

HALT   #005H }-----⑥ HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV    T, #TIME9M }-----⑦ 0個の場合, 計18ms送信するためにタイマ・カウンタに再度9ms (=1FFH) を設定
                               します。

T9_UNDER:
STTS   #0101B }-----⑧ ステータス・フラグをクリアします。
MOV    A, R09 }-----⑨ データ1送信カウンタL (R09) の最上位ビットをクリアします。
ANL   A, #0111B }-----
MOV    R00, A }-----⑩ データ1送信カウンタL (R09) の値をデータ・ポインタL (R00) に設定しま
                               す。
HALT   #005H }-----⑪ HALTモード: タイマ・カウンタが00Hになると解除されます。
MOV    T, @R0 }-----⑫ テーブル参照により, タイマ・カウンタにビット・データ '1' の数に応じて
                               0.940ms~8.815msを設定します。
STTS   #0101B }-----⑬ ステータス・フラグをクリアします。

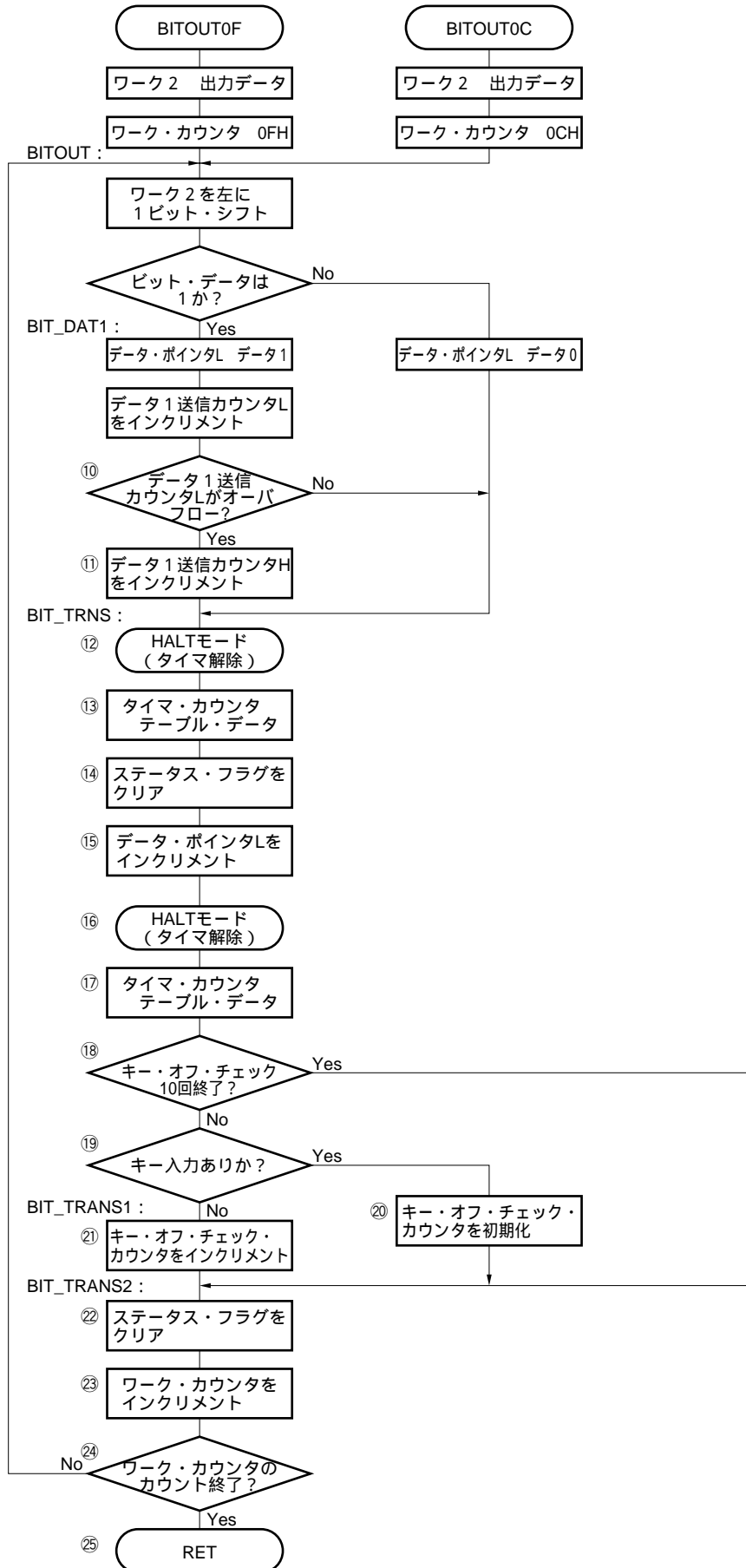
MOV    A, R03 }-----⑭ 1フレーム目の送信 (連続押しではないか) を判断します。
SCAF   }-----      1フレーム目の送信の場合, KEY_ONへ
JNC   KEY_ON }-----
MOV    A, R07 }-----⑮ 2フレーム目以降の送信 (連続押し) の場合, 送信中にキー・オフになったかを
SCAF   }-----      判断します。
JNC   KEY80 }-----      連続押しの場合, KEY80へ

; ** Frame Space = 27ms **
CALL   FS_27MS }-----⑯ ビット・データ送信中にキー・オフになった場合, フレーム・スペースの残り
                               27ms出力します。
HALT   #005H }-----⑰ HALTモード: タイマ・カウンタが00Hになると解除されます。
JMP    MAIN }-----      初期設定処理, MAINへ
;

KEY_ON:
MOV    A, R17 }-----⑱ 連続フラグ (R03) をセットします (このときR17=0FH)。
MOV    R03, A }-----

; ** Frame Space = 27ms **
CALL   FS_27MS }-----⑲ 27ms出力用サブルーチンをコールします。
JMP    TRNS }-----      TRNSへ
;

```

```

;*****
;****          ****
;****          Transmit          ****
;****          Subroutine : Bit Out      ****
;****          ****
;*****
BITOUTOF:
    MOV    R1A, A                ① ワーク2 (R1A) に出力データを設定します。
    MOV    A, #0FH              ② 送信ビット・カウンタを1回に設定するためにアキュムレータを0FHに設定
                                します。
    JMP    BITOUT                BITOUTへ
;
BITOUTOC:
    MOV    R1A, A                ③ ワーク2 (R1A) に出力データを設定します。
    MOV    A, #0CH              ④ 送信ビット・カウンタを4回に設定するためにアキュムレータを0CHに設定
                                します。
BITOUT:
    MOV    R0A, A                [備考] 送信ビット・カウンタ (R0A) に0FH, 0EH, 0CHのいずれかを設定します。
    MOV    A, R1A
    RL     A                      ⑤ 出力データを左へ1ビット・シフトします。
    MOV    R1A, A
    JC     BIT_DAT1              ⑥ ビット・データが '1' か '0' かを判断します。
                                '1' の場合、BIT_DAT1へ
;
; ** Bit Data = 0 **
    MOV    A, R08
    MOV    R00, A                ⑦ データ・ポインタL (R00) にデータ0 (R08) を設定します。
    JMP    BIT_TRN                BIT_TRNSへ
;
; ** Bit Data = 1 **;
BIT_DAT1:
    MOV    A, R18
    MOV    R00, A                ⑧ データ・ポインタL (R00) にデータ1 (R18) を設定します。
    MOV    A, R09
    INC    A                      ⑨ データ1送信カウンタL (R09) をインクリメントします。
    MOV    R09, A
    JNC    BIT_TRANS              ⑩ データ1送信カウンタL (R09) がオーバーフローしなければ、BIT_TRNSへ
    MOV    A, R19
    INC    A                      ⑪ データ1送信カウンタH (R19) をインクリメントします。
    MOV    R19, A
BIT_TRNS:
    HALT   #005H                  ⑫ HALTモード：タイマ・カウンタが00Hになると解除されます。
    MOV    T, @R0                  ⑬ テーブル参照により、タイマ・カウンタに送信時間データを設定します。
    STTS   #0101B                  ⑭ ステータス・フラグをクリアします。
    MOV    A, R00
    INC    A                      ⑮ データ・ポインタL (R00) をインクリメントします。
    MOV    R00, A
    HALT   #005H                  ⑯ HALTモード：タイマ・カウンタが00Hになると解除されます。
    MOV    T, @R0                  ⑰ テーブル参照により、タイマ・カウンタに送信時間データを設定します。
    MOV    A, R07
    INC    A                      ⑱ 10回連続してキー・オフが確認したかを判断します。
    JC     BIT_TRANS2              キー・オフ確定の場合、BIT_TRANS2へ
    STTS   #1110B                  ⑲ キー入力あり、かどうかを判断します。
    JNF    BIT_TRANS1              キーが押し続けの場合、BIT_TRANS1へ
;
; ** キー入力あり **
    MOV    R7, #0F5H              ⑳ キー・オフ・チェック (10回) 中に1回でもキー入力があった場合は、キー・
                                オフ・チェック・カウンタ (R07) をイニシャライズ (=5H) します。
    JMP    BIT_TRANS2              BIT_TRANS2へ
;

```

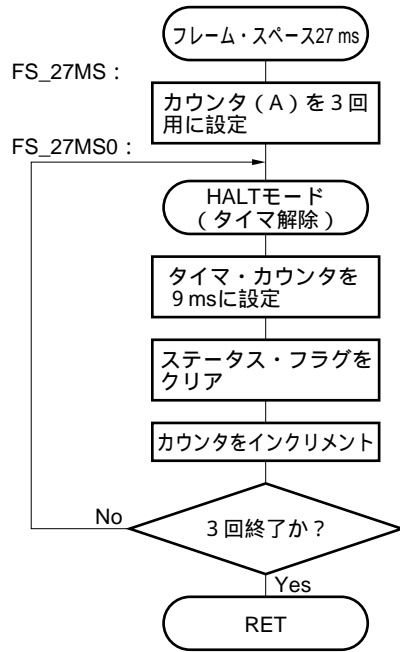
```

    ;** キー入力なし **
BIT_TRANS1:
    MOV     R07, A
BIT_TRANS2:
    STTS   #0101B
    MOV    A, R0A
    INC    A
    JNC    BITOUT

    RET
    ;
END

```

㉑ キー・オフ・チェック・カウンタをインクリメントします。
 ㉒ ステータス・フラグをクリアします。
 ㉓ 送信ビット・カウンタ (R0A) をインクリメントします。
 ㉔ 送信ビット数のカウントを終了したかを判断します。
 終了していない場合は、BITOUTへ
 ㉕ 終了



```

;*****
;*****
;*****      FRAME SPACE = 27ms      ***** [備考] フレーム・スペースの残り27ms出力用サブルーチン
;*****      Subroutine : FS 27ms      *****
;*****
;*****
FS_27MS:
    MOV     A, #0DH                    ① カウンタ（アキュムレータ）を3回（=0DH）に設定します。
FS_27MS0:
    HALT   #005H                      ② HALTモード：タイマ・カウンタが00Hになると解除されます。
    MOV    T, #TIME9M                 ③ タイマ・カウンタを9ms（=1FFH）に設定します。
    STIS   #0101B                    ④ ステータス・フラグをクリアします。
    INC    A                          ⑤ カウンタ（アキュムレータ）をインクリメントします。
    JNC    FS_27MS0                  ⑥ 3回カウント終了したかを判断します。
                                           終了していない場合、FS_27MS0へ
    RET                                ⑦ 終了
END

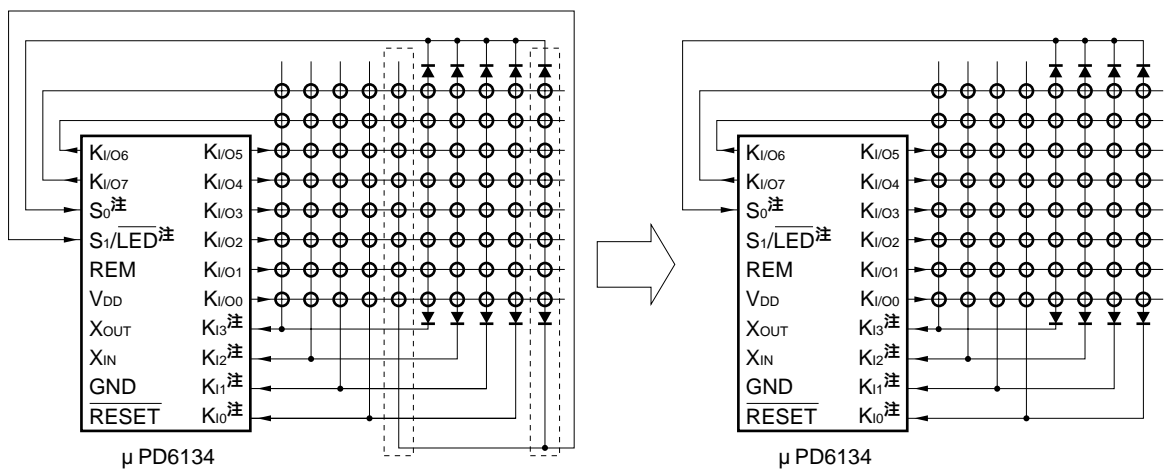
```

第7章 プログラム変更時の注意点

キー・マトリクスおよびキー数の変更を行う際には、次の点に注意してください。

(1) 80キーから64キーに変更する場合

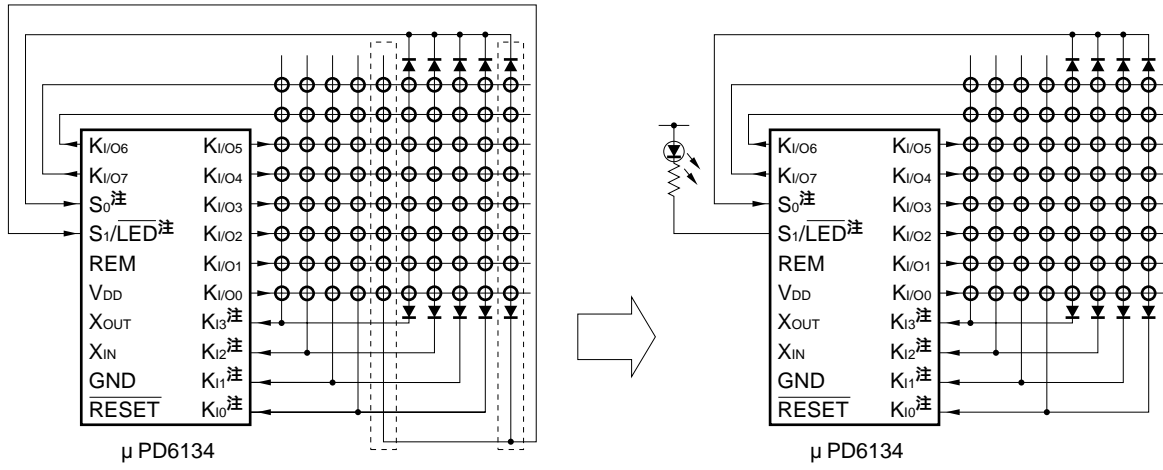
- ・次の図の破線で囲まれたキーを削除してください（そのとき、 $\overline{S_1/LED}$ 端子をオープンにすることによって、本プログラムの内容を変更せずにキー数を変更することができます）。



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

(2) (1)の変更に加えて、 $\overline{S_1/LED}$ 端子をLEDとして使用する場合

- ・次の図の破線で囲まれたキーを削除してください。
- ・ $\overline{S_1/LED}$ 端子は本プログラム内でコントロール・レジスタ1 (P4) のb2 ($\overline{S_1/LED}$ ポートの入力モード設定用のビット) を常に出力モード (=1) に設定するように修正してください (μPD6133シリーズでは、 $\overline{S_1/LED}$ 端子を出力モードにすることによって、内蔵プルダウン抵抗は自動的に切れます)。



注 プログラムで“プルダウン抵抗を内蔵する”ように設定しています。

第 8 章 本プログラム使用上の注意点

本プログラムのキーの二重押しの仕様では、K29 + K30, K29 + K31, K29 + K32の3パターンのみ有効となっています。またキー・マトリクスの構成上、同一キー・ソース・ライン上の通常キー（K1-K40）と拡張キー（K41-K80）とを二重押しすると、常に拡張キーが有効になります。

たとえば、K19 + K59の二重押しの場合、K59が有効になります。

〔メ モ〕

第9章 プログラム・リスト

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:00 08/29/96 PAGE 01-002

SOURCE = MAIN80.ASM

E STNO	LOC.	OBJ.	M I	SOURCE STATEMENT
1				*****
2				***
3				*** Multi - Purpose Remote Control Transmitter System ***
4				***
5				*** CPU : uPD6133 Series ***
6				*** CPU Clock : 455kHz ***
7				*** Trans. CODE: NEC-R Format (80Key) ***
8				*** Version : 2.0 ***
9				*** Programmer : NEC IC Microcomputer Systems Corporation ***
10				***
11				*** Copyright(c) NEC Corporation 1995 ***
12				*** Copyright(c) NIMS Corporation 1995 ***
13				*****
14				EJECT

SOURCE = MAIN80.ASM

```

E STNO LOC. OBJ.      M I SOURCE STATEMENT
15 001C                TIME05M      EQU      01CH                ;0.5ms (0.510ms)
16 01FF                DmyTime     EQU      512-1          ;9.00ms (= 17.582us * 512)
17
18                    ;##### P U B L I C #####
19                    PUBLIC      MAIN
20                    ;
21                    ;##### E X T E R N #####
22                    ;
23
24                    ;##### S T A R T #####
25
26                    ; Control Register (P3)      Reset:03H
27                    ;=====;
28                    ; D9 D8 ! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 ! ;
29                    ;          ! ! D.P.! D.P.! D.P.! TCTL! CARY! MOD1! MOD0! ;
30                    ; Test Mode! ! AD10! AD9 ! AD8 ! ! ! ! ! ;
31                    ;-----;
32                    ;          ! 0 ! * ! * ! * ! 1/1 ! ON ! fx,fx/8 ! 0 ;
33                    ; Set "0" !-----! ,fx/12(1/2)!----;
34                    ;          ! 0 ! * ! * ! * ! 1/2 ! OFF ! ,fx/12(1/3)! 1 ;
35                    ;=====;
36
37                    ; Control Register (P4)      Reset:26H
38                    ;=====;
39                    ; D9 D8 ! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 ! ;
40                    ;          ! ! ! KI !S0/S1! !S1/LED! KI/O! S0 ! ;
41                    ;          ! ! ! Pull! Pull! ! MODE ! MODE! MODE! ;
42                    ;-----;
43                    ;          ! 0 ! 0 ! OFF ! OFF ! 0 ! IN ! IN ! OFF ! 0 ;
44                    ; X !-----!-----!-----;
45                    ;          ! 0 ! 0 ! ON ! ON ! 0 ! OUT ! OUT ! IN ! 1 ;
46                    ;=====;
47
48                    ;#####
49                    ; Control Register Initialize
50                    ;#####
51                    MAIN:
52 0000 E6F8 EFEF                OUT      P0,#0FFH                ;KI/O All High
53 0002 E6FC E3E2                OUT      P4,#032H
54 0004 E6FB E1E3                OUT      P3,#013H                ;Set Data Pointer (P3:D8,D9)
55 0006 E6FF F7FF                MOV      T,#DmyTime
56 0008 E3F1 E0E5                STTS     #0101B                ;Clear Status Flag
57
58                    ;#####
59                    ; RAM Initialize Routine
60                    ;#####
61 000A E6E1 E0E0                MOV      R1,#000H                ;Final Key Data (R1) = 00H
62 000C E6E3 E0E0                MOV      R3,#000H                ;K29 ON Flag (R13) , Continuance Flag (R03) = 00H
63
64 000E E2F1 E0E8                HALT     #008H                ;STOP mode (KI = High)
65 0010 E0E0                NOP                                ;No operation command
66 0011 E6FF E0E7                MOV      T,#TIME05M            ;Set Timer : 0.5ms
67                    ;
68                    END

```

SOURCE = KEY80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
1                          ;*****
2                          ;*****
3                          ;*****      uPD6133 Series Key Check      *****
4                          ;*****      Trans. CODE: NEC-R Format (80Key) *****
5                          ;*****
6                          ;*****
7      01FF              TIME9M EQU  1FFH                      ;9.00ms (9.002ms)
8      0050              CUSTM1 EQU  050H                      ;Custom Code = 0AH
9      00AF              CUSTM2 EQU  0AFH                      ;Custom Code' = F5H
10
11                      ;##### P U B L I C #####
12                      PUBLIC      KEY80
13                      ;
14
15                      ;##### E X T E R N #####
16                      EXTRN      MAIN                          ;MAIN Routine
17                      ;
18
19                      ;##### S T A R T #####
20                      KEY80:
21      0013 E6E0 E0EC          MOV   R0,#00CH                      ;Chattering Counter (R0) = 3 Times
22      0015 E6EC E8E2          MOV   RC,#082H                      ;Set Immediate Data : RC
23
24                      ;*****
25                      ;           ON Chattering
26                      ;*****
27      0017 E3F1 E0E5          STTS  #0101B                          ;Clear Status Flag
28      0019 E6F8 EFEF          OUT  P0,#0FFH                          ;KI/O All High
29      001B E6FC E3E3          OUT  P4,#033H
30      001D E2F1 E0E5          HALT #005H                          ;HALT mode (Timer = 00H)
31      001F E6FF F7FF          MOV  T,#TIME9M                          ;Timer = 9.00ms
32      0021 E6E9 E0E0          MOV  R9,#000H                          ;Comparative Key Data (R9) = 00H
33      0023 E6E6 E0E8          MOV  R6,#008H                          ;Key Flag (R16) = 00H , Key Scan Counter (R06) = 08H
34      0025 E3F1 E0EE          STTS #1110B
35      0027 EFF1 E0E0          JNF  MAIN                               ;   else MAIN: (Without input Key)
36
37      0029 E6E8 E0E1          MOV  R8,#001H                          ;   then      (With input Key)
38
39
40                      ;*****
41                      ;           Key Scan
42                      ;*****
43      002B FFE8              MOV   A,R08
44      002C E5F8              OUT  P00,A                             ;Key Scan Data L (P00) Output
45      002D FEE8              MOV  A,R18
46      002E E4F8              OUT  P10,A                             ;Key Scan Data H (P10) Output
47
48                      ;*****
49                      ;++      100us Wait      ++
50      002F E6E2 EFE1          MOV  R2,#0F1H                          ;Set Immediate Data : R2
51      0031 E6EB ECE3          MOV  RB,#0C3H                          ;Set Immediate Data : RB
52      0033 E0E0              NOP
53      0034 E0E0              NOP
54      0035 E0E0              NOP
55      0036 E0E0              NOP

```

SOURCE = KEY80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56                      ;*****
57                      ;           Key Return Check
58                      ;*****
59 0037 E3F1 E0EB          STTS  #1011B          ;With input Key (KI or S0 or S1) ?
60 0039 EFF1 E9E8          JNF   NO_KEYS          ;   else NO_KEYS: (Without input Key)
61                      ;   then          (With input Key)
62 003B E3F1 E0E3          STTS  #0011B          ;With input KI (P11) ?
63 003D EFF1 E4E8          JNF   SOS1CALC        ;   else SOS1CALC: (With input KI)
64                      ;   then          (Without input KI)
65                      ;*****;   then          (KI(P11)入力なし)
66                      ;   KI Data Calculate
67                      ;*****
68 003F FFF9              IN    A,P01
69 0040 FCF3              RL    A                ;With input S1(P01) ?
70 0041 ECF1 E0E0          JC    MAIN            ;   else MAIN: (With input S1)
71                      ;   then          (Without input S1)
72 0043 FEEB              MOV   A,R1B           ;R1B = 0CH
73 0044 E5EA              MOV   ROA,A           ;Key Return Check Counter (ROA) = 4 Times (=0CH)
74 0045 FEF9              IN    A,P11
75 0046 E8F1 E4ED          JMP   KEY_RET0
76                      ;
77
78                      ;*****
79                      ;   S0,S1 Data Calculate
80                      ;*****
81
82 0048 E6E9 EBEO          MOV   R9,#0BOH
83 004A FEE2              MOV   A,R12           ;R12 = 0FH
84 004B E5EA              MOV   ROA,A           ;Set Key Return Check Counter (ROA) = 1 Times (=0FH)
85 004C FFF9              IN    A,P01
86
87
88 004D E4E0          KEY_RET0:            MOV   R10,A           ;Work (R10) = P11 or P01
89 004E E3F1 E0E5          STTS  #0101B         ;Clear Status Flag
90
91 0050 FEE0          KEY_RET:            MOV   A,R10
92 0051 FEF3              RLZ   A
93 0052 E4E0          MOV   R10,A
94 0053 EDF1 E9E3          JNC   KRC_INC        ;With input Key ?
95                      ;** With input Key ** ;   else KRC_INC: (Without input Key)
96 0055 FEE6              MOV   A,R16           ;   then          (With input Key)
97 0056 FAF3              SCAF          ;Key Flag (R16) = 0FH ?
98 0057 EDF1 E6E7          JNC   KEY_CALC       ;   else KEY_CALC: (Except for 0FH)
99 0059 FEE3              MOV   A,R13           ;   then          (= 0FH)
100 005A FAF3              SCAF          ;K29 ON Flag (R13) = 0FH ?
101 005B EDF1 E0E0          JNC   MAIN           ;   else MAIN: (= Except for 0FH)
102                      ;   then          (= 0FH)
103 005D FFEC              MOV   A,ROC           ;ROC = 02H
104 005E F4E9              XRL   A,R19
105 005F FAF3              SCAF          ;Comparative Key Data = K29 (R19=0DH) ?
106 0060 EDF1 E0E0          JNC   MAIN           ;   then MAIN: (Key = Except for K29)
107 0062 FFE9              MOV   A,RO9           ;   else          (Key = K29)
108 0063 F5EB              XRL   A,ROB           ;ROB = 03H
109 0064 FAF3              SCAF          ;Comparative Key Data = K29 (RO9=0CH) ?
110 0065 EDF1 E0E0          JNC   MAIN           ;   then DBL_K_CHK: (Key = Except for K29)

```

SOURCE = KEY80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111
112                      ;*****
113                      ;   Key Data Calculate
114                      ;*****
115          KEY_CALC:
116 0067 FFF1 F4E0          MOV   A,#04H
117 0069 F4E9              XRL   A,R19
118 006A FAF3              SCAF                ;Comparative Key Data H (R19) = 0BH ?
119 006B EDF1 E7E5          JNC   KI_CALC                ;   else KI_CALC: (KI Data)
120                      ;** S1 data Calculation ** ;   then      (S1 Data)
121 006D FFE6              MOV   A,R06                ;Acc <-- Key Scan Counter (R06)
122 006E FBF1 F7E0          ANL   A,#0111B
123 0070 E5E9              MOV   R09,A
124 0071 FEE0              MOV   A,R10                ;Acc <-- Work 1 (R10)
125 0072 FAEC              ANL   A,R1C                ;R1C = 08H
126 0073 E8F1 E8E8          JMP   KL_CALC
127                      ;
128                      ;** KI data Calculation **
129          KI_CALC:
130 0075 FFE6              MOV   A,R06                ;Acc <-- Key Scan Counter(R06)
131 0076 FCF3              RL    A
132 0077 FCF3              RL    A                ;Left shift (2 Times)
133 0078 E5E9              MOV   R09,A                ;Comparative Key Data L (R09) <-- Acc
134 0079 FBE2              ANL   A,R02                ;Acc AND 0001B (R02=01H)
135 007A FCEB              ORL   A,R1B                ;Acc OR 1100B (R1B=0CH)
136 007B E4E9              MOV   R19,A                ;Comparative Key Data H (R19) <-- Acc
137 007C FFF9              IN   A,P01                ;Acc <-- S1 and S0 Data
138 007D FCF3              RL    A                ;Left shift (3 Times)
139 007E FCF3              RL    A
140 007F FCF3              RL    A
141 0080 FBEC              ANL   A,R0C                ;Acc AND 0010B (R0C=02H)
142 0081 FCE9              ORL   A,R19                ;Acc OR Comparative Key Data H (R19)
143 0082 E4E9              MOV   R19,A                ;Comparative Key Data H (R19) <-- Acc
144 0083 FFE9              MOV   A,R09                ;Acc <-- Comparative Key Data L (R09)
145 0084 FAEB              ANL   A,R1B                ;Acc AND 1100B (R1B=0CH)
146 0085 E5E9              MOV   R09,A                ;Comparative Key Data L (R09) <-- Acc
147 0086 FFEA              MOV   A,ROA                ;Acc <-- Key Return Check Counter (ROA)
148 0087 FBEB              ANL   A,ROB                ;Acc AND 0011B (ROB=03H)
149          KI_CALC:
150 0088 FDE9              ORL   A,R09                ;Acc OR Comparative Key Data L (R09)
151 0089 E5E9              MOV   R09,A                ;Comparative Key Data L (R09) <-- Acc
152
153                      ;** Double Key Check **
154 008A FEE6              MOV   A,R16
155 008B FAF3              SCAF
156 008C EDF1 E9E1          JNC   SINGLE_K                ;Key Flag (R16) = 0FH ?
157 008E FFF1 F9E0          MOV   A,#09H                ;   else SINGLE_K (Single Key)
158 0090 E4E9              MOV   R19,A                ;   then      (Double Key)
159          SINGLE_K:
160 0091 FEE2              MOV   A,R12                ;R12 = 0FH
161 0092 E4E6              MOV   R16,A
162
163          KRC_INC:
164 0093 FFEA              MOV   A,ROA
165 0094 F4F3              INC   A                ;Key Return Check Counter (ROA) increment

```

SOURCE = KEY80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
166 0095 E5EA              MOV    R0A,A                ;End of Key Return Check ?
167 0096 EDF1 E5E0        JNC    KEY_RET              ;   else KEY_RET: (Continuous)
168                          ;   then          (End)
169
170                          NO_KEYS:
171 0098 E3F1 E0E5        STTS   #0101B              ;Clear Status Flag
172 009A FFE6              MOV    A,R06
173 009B F4F3              INC    A                    ;Key Scan Counter(R06) increment
174 009C E5E6              MOV    R06,A                ;End of Key Scan number of 8 Time ?
175 009D ECF1 EAEB        JC     KEY_DATA            ;   then KEY_DATA: (End)
176 009F FEE8              MOV    A,R18                ;   else          (Continuous)
177 00A0 FCF3              RL     A                    ;Key Scan Data H (R18) Shift
178 00A1 E4E8              MOV    R18,A
179 00A2 FFE8              MOV    A,R08
180 00A3 FCF3              RL     A
181 00A4 E5E8              MOV    R08,A                ;Key Scan Data L (R08) Shift --> CY ?
182 00A5 EDF1 E2EB        JNC    KEYS_OUT            ;   else KEYS_OUT: (Key Scan Data a low rank)
183 00A7 EGE8 E1E0        MOV    R8,#010H           ;   then          (Key Scan Data a high rank)
184 00A9 E8F1 E2EB        JMP    KEYS_OUT
185                          ;
186
187                          ;*****
188                          ; Transmit Key Data = ON Chattering Key Data ?
189                          ;*****
190                          KEY_DATA:
191 00AB E6F8 EFEF        OUT    P0,#0FFH           ;KI/O All High
192 00AD EGE8 EFE0        MOV    R8,#0FOH           ;Set Immediate Data : R8
193 00AF FEE6              MOV    A,R16
194 00B0 FAF3              SCAF
195 00B1 EDF1 E0E0        JNC    MAIN                ;Key Flag (R16) = 0FH ?
196 00B3 FFE0              MOV    A,R00                ;   else MAIN:   (= Except for 0FH)
197 00B4 F5EB              XRL   A,R0B                ;   then          (= 0FH)
198 00B5 FAF3              SCAF                        ;R0B = #0011B
199 00B6 ECF1 ECE3        JC     KEYCHECK            ;Chattering Counter (R00) = 1 Times (=0DH) ?
200                          ;   then KEYCHECK: (= 0DH)
201 00B8 FEE4              MOV    A,R14                ;   else          (= 0E-0FH)
202 00B9 F4E8              XRL   A,R18                ;R18 = 0FH
203 00BA F4E9              XRL   A,R19                ;Acc = 0FH
204 00BB FAF3              SCAF                        ;Key Data H (R14) = Comparative Key Data H (R19)
205 00BC EDF1 E0E0        JNC    MAIN                ;   else MAIN:   (Unmatch)
206                          ;   then          (Match)
207 00BE F5E4              XRL   A,R04                ;Acc = 0FH
208 00BF F5E9              XRL   A,R09
209 00C0 FAF3              SCAF                        ;Key Data L (R04) = Comparative Key Data L (R09)
210 00C1 EDF1 E0E0        JNC    MAIN                ;   else MAIN:   (Unmatch)
211                          ;   then          (Match)
212                          KEYCHECK:
212 00C3 FEE9              MOV    A,R19
213 00C4 E4E4              MOV    R14,A                ;Key Data H (R14) ← Comparative Key Data H (R19)
214 00C5 FFE9              MOV    A,R09
215 00C6 E5E4              MOV    R04,A                ;Key Data L (R04) ← Comparative Key Data L (R09)
216 00C7 FFE0              MOV    A,R00
217 00C8 F4F3              INC    A                    ;Chattering Counter (R00) increment
218 00C9 E5E0              MOV    R00,A
219 00CA FAF3              SCAF
220 00CB EDF1 E1E7        JNC    ON_CHAT            ;End of Chattering Routine of 3 Time ?
                          ;   else ON_CHAT: (= 0DH)

```

SOURCE = KEY80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
221                      ;** Key High Check **          ;   then          (= 0E-0FH)
222 00CD F4E1             XRL   A,R11                ;Acc = 0FH
223 00CE F4E4             XRL   A,R14
224 00CF FAF3             SCAF
225 00D0 EDF1 EDE7       JNC   KDAT_SET                ;Final Key Data H (R11) = Key Data H (R14) ?
226                      ;   else KDAT_SET: (Unmatch)
227 00D2 F5E1             ;** Key Low Check **          ;   then          (Match)
228 00D3 F5E4             XRL   A,R01                ;Acc = 0FH
229 00D4 FAF3             XRL   A,R04
230 00D5 ECF1 EDED       SCAF
231                      JC    K29CHECK                ;Final Key Data L(R01) = Key Data L (R04) ?
232 00D7 FEE4             MOV   A,R14                ;   then K29CHECK: (Match)
233 00D8 E4E1             MOV   R11,A                ;   else          (Unmatch)
234 00D9 FFE4             MOV   A,R04
235 00DA E5E1             MOV   R01,A                ;Final Key Data H (R11) <-- Key Data H (R14)
236 00DB FFE8             MOV   A,R08                ;Final Key Data L (R01) <-- Key Data L (R04)
237 00DC E5E3             MOV   R03,A                ;Continuance Flag = 00H (R08=00H)
238
239                      K29CHECK:
240 00DD FFEC             MOV   A,ROC                ;ROC = 02H (K29=1DCH)
241 00DE F4E1             XRL   A,R11
242 00DF FAF3             SCAF
243 00E0 EDF1 EEE7       JNC   DBL_K_CHK                ;Final Key Data = K29 (R11=0BH) ?
244 00E2 FFE1             MOV   A,R01                ;   then DBL_K_CHK: (Key = Except for K29)
245 00E3 F5EB             XRL   A,ROB                ;   else          (Key = K29)
246 00E4 FAF3             SCAF
247 00E5 ECF1 EEEE       JC    K29_FLG                ;ROB = 03H
248                      ;Final Key Data = K29 (R01=0CH) ?
249                      ;   then K29_FLG: (Key = Except for K29)
250 00E7 FFF1 F6E0       MOV   A,#06H                ;   else          (Key = K29)
251 00E9 F4E1             XRL   A,R11
252 00EA FAF3             SCAF
253 00EB ECF1 EEEF       JC    DATA_SET                ;Final Key Data = K29+ K30 or K31 or K32 (R11=0BH) ?
254 00ED FFE8             MOV   A,R08                ;   then DATA_SET: (Key = Except for Double Key)
255                      K29_FLG:
256 00EE E4E3             MOV   R13,A                ;K29 ON Flag = 00H (R08=00H)
257                      ;*****
258                      ;   Transmit Data Set
259                      ;*****
260                      DATA_SET:
261                      ;** Custom Code **
262 00EF E6E4 E5E0       MOV   R4,#CUSTM1                ;Custom Code(R4) <-- Custom Code
263
264                      ;** Custom Code' **
265 00F1 E6E5 EAEF       MOV   R5,#CUSTM2                ;Custom Code'(R5) <-- Custom Code'
266
267                      ;** Data Code **
268 00F3 FEE1             MOV   A,R11
269 00F4 E4E0             MOV   R10,A                ;Data Pointer H(R10) <-- Final Key Data H(R11)
270 00F5 FFE1             MOV   A,R01
271 00F6 E5E0             MOV   R00,A                ;Data Pointer L(R00) <-- Final Key Data L(R01)
272 00F7 E7E6             MOV   R6,#R0
273
274                      END

```


SOURCE = TRANS80.ASM

```

E SINO LOC. OBJ.      M I SOURCE STATEMENT
1                    ;*****
2                    ;*****          *****
3                    ;*****      uPD6133 Series Transmit      *****
4                    ;*****      Trans. CODE: NEC-R Format (80Key) *****
5                    ;*****          *****
6                    ;*****
7      01FF          TIME9M      EQU      1FFH          ;9.00ms (9.002ms)
8
9                    ;##### P U B L I C #####
10                   ;
11
12                   ;##### E X T E R N #####
13                   EXTRN      MAIN          ;MAIN Routine
14                   EXTRN      KEY80        ;Key Check Routine
15                   EXTRN      FS_27MS      ;Frame Space Subroutine
16                   ;
17
18                   ;##### S T A R T #####
19                   TRNS:
20      00F8 E6E7 EFE5      MOV      R7,#0F5H          ;R17 : Set Immediate Data (=0FH)
21
22      00FA E6E9 EDEF      MOV      R9,#0DFH          ;Key OFF Check Counter (R07) = 10 Times (=05H)
23                               ;Data Trans Counter (R9) = DFH
24                   ; Leader Code
25                   ;*****
26      00FC E6E0 EAE0      MOV      R0,#0A0H          ;Data Pointer H (R10) = 0AH
27      00FE E6E8 E0E8      MOV      R8,#008H          ;Set Table Address L
28      0100 FEE8          MOV      A,R18          ;Output Data (Work) <-- 00H (R18)
29      0101 E6F2 E8F1 E6F1      CALL   BITOUTOF
30
31                   ;*****
32                   ; Custom Code
33                   ;*****
34      0104 E6E8 ECEA      MOV      R8,#0CAH          ;Set Table Address L
35      0106 FEE4          MOV      A,R14          ;Output Data (Work) <-- Custom Code H (R14)
36      0107 E6F2 E8F1 E6F6      CALL   BITOUTOC
37
38      010A FFE4          MOV      A,R04          ;Output Data (Work) <-- Custom Code L (R04)
39      010B E6F2 E8F1 E6F6      CALL   BITOUTOC
40
41                   ;*****
42                   ; Custom Code'
43                   ;*****
44      010E FEE5          MOV      A,R15          ;Output Data (Work) <-- Custom Code' H (R15)
45      010F E6F2 E8F1 E6F6      CALL   BITOUTOC
46
47      0112 FFE5          MOV      A,R05          ;Output Data (Work) <-- Custom Code' L (R05)
48      0113 E6F2 E8F1 E6F6      CALL   BITOUTOC
49
50                   ;*****
51                   ; Data Code
52                   ;*****
53      0116 FEE6          MOV      A,R16          ;Output Data (Work) <-- Data Code H (R16)
54      0117 E6F2 E8F1 E6F6      CALL   BITOUTOC
55

```

SOURCE = TRANS80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56 011A FFE6              MOV    A,R06                ;Output Data (Work) <-- Data Code L (R06)
57 011B E6F2 E8F1 E6F6    CALL  BITOUTOC
58
59
60                        ;*****
61                        ;          Data Code'
62 011E FEE6              MOV    A,R16                ;Output Data (Work) <-- Data Code H (R16)
63 011F F4E7              XRL   A,R17                ;R17 = 0FH
64 0120 E6F2 E8F1 E6F6    CALL  BITOUTOC
65
66 0123 FFE6              MOV    A,R06                ;Output Data (Work) <-- Data Code L (R06)
67 0124 F4E7              XRL   A,R17                ;R17 = 0FH
68 0125 E6F2 E8F1 E6F6    CALL  BITOUTOC
69
70                        ;*****
71                        ;          Stop Bit
72                        ;*****
73 0128 E6E8 E0EE          MOV    R8,#00EH            ;Set Table Address L
74 012A FEE8              MOV    A,R18                ;Output Data (Work) <-- 00H (R18)
75 012B E6F2 E8F1 E6F1    CALL  BITOUTOF
76
77                        ;*****
78                        ;          Frame Space
79                        ;*****
80 012E FEE9              MOV    A,R19
81 012F FAF3              SCAF
82 0130 ECF1 E4F0          JC     T9_UNDER            ;Frame Space > 9.00ms ?
83                        ;   then T9_UNDER: (More than 9.00ms)
84 0132 E2F1 E0E5          HALT  #005H                ;   else (Less than 9.00ms)
85 0134 E6FF F7FF          MOV    T,#TIME9M           ;HALT mode (Timer = 00H)
86 0136 E3F1 E0E5          STTS  #0101B              ;Clear Status Flag
87
88 0138 FFE9              MOV    A,R09
89 0139 FCF3              RL    A                    ;Frame Space > 18.00ms ?
90 013A ECF1 E4F0          JC     T9_UNDER            ;   then T9_UNDER: (Less than 18.00ms)
91                        ;   else (More than 18.00ms)
92 013C E2F1 E0E5          HALT  #005H                ;HALT mode (Timer = 00H)
93 013E E6FF F7FF          MOV    T,#TIME9M
94
95 T9_UNDER:
96 0140 E3F1 E0E5          STTS  #0101B              ;Clear Status Flag
97 0142 FFE9              MOV    A,R09
98 0143 FBF1 F7E0          ANL   A,#0111B
99 0145 E5E0              MOV    R00,A
100 0146 E2F1 E0E5         HALT  #005H                ;HALT mode (Timer = 00H)
101 0148 E7FF              MOV    T,R0
102 0149 E3F1 E0E5         STTS  #0101B              ;Clear Status Flag
103
104 014B FFE3              MOV    A,R03
105 014C FAF3              SCAF
106 014D EDF1 E5FA          JNC   KEY_ON              ;Continuously pressed key ?
107 014F FFE7              MOV    A,R07                ;   else KEY_ON: (1st. Frame)
108 0150 FAF3              SCAF                        ;   then (Since 2nd. Frame)
109 0151 EDF1 E1E3          JNC   KEY80                ;Without input Key ?
110                        ;   then KEY80: (With input Key)
                        ;   else (Without input Key)

```

SOURCE = TRANS80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111 0153 EXTRN                CALL  FS_27MS
112 0156 E2F1 E0E5            HALT  #005H          ;HALT mode (Timer = 00H)
113 0158 E8F1 E0E0            JMP   MAIN
114                            ;
115
116 015A FEE7                MOV   A,R17
117 015B E5E3                MOV   R03,A          ;Continuance Flag (R03) = 0FH (R17=0FH)
118                            ;** Frame Space = 27ms **
119 015C EXTRN                CALL  FS_27MS
120 015F E8F1 EFE8            JMP   TRNS
121                            ;
122
123                            ;*****
124                            ;****              ****
125                            ;**** Transmit          ****
126                            ;**** Subroutine : Bit Out ****
127                            ;****              ****
128                            ;*****
129 BITOUTOF:
130 0161 E4EA                MOV   R1A,A          ;Set Output Data (1 Bit Output)
131 0162 FFF1 FFE0            MOV   A,#0FH          ;Send Bit Counter = 1 Times (=0FH)
132 0164 E8F1 E6F9            JMP   BITOUT
133                            ;
134 BITOUTOC:
135 0166 E4EA                MOV   R1A,A          ;Set Output Data (4 Bit Output)
136 0167 FFF1 FCE0            MOV   A,#0CH          ;Send Bit Counter = 4 Times (=0CH)
137
138 BITOUT:
139 0169 E5EA                MOV   R0A,A          ;Send Bit Counter (R0A) = 0CH or 0EH or 0FH
140
141 016A FEEA                MOV   A,R1A
142 016B FCF3                RL   A
143 016C E4EA                MOV   R1A,A          ;Bit Data = 1 ?
144 016D ECF1 E7F3            JC   BIT_DAT1        ; then BIT_DAT1: (Bit Data = 1)
145                            ;** Bit Data = 0 ** ; else (Bit Data = 0)
146 016F FFE8                MOV   A,R08          ;Set Data Pointer L (R00) <-- Data0 (R08)
147 0170 E5E0                MOV   R00,A
148 0171 E8F1 E7FD            JMP   BIT_TRNS
149                            ;
150                            ;** Bit Data = 1 **;
151 BIT_DAT1:
152 0173 FEE8                MOV   A,R18          ;Set Data Pointer L(R00) <-- Data1 (R18)
153 0174 E5E0                MOV   R00,A
154 0175 FFE9                MOV   A,R09
155 0176 F4F3                INC  A               ;Data 1 Transmit Counter L (R09) increment
156 0177 E5E9                MOV   R09,A          ;Data 1 Transmit Counter L (R09) = Overflow ?
157 0178 EDF1 E7FD            JNC  BIT_TRNS        ; else BIT_TRNS: (Data 1 Transmit Counter L <= 0FH)
158 017A FEE9                MOV   A,R19          ; then (Data 1 Transmit Counter L > 0FH)
159 017B F4F3                INC  A               ;Data 1 Transmit Counter H(R19) increment
160 017C E4E9                MOV   R19,A
161
162 017D E2F1 E0E5            HALT  #005H          ;HALT mode (Timer = 00H)
163 017F E7FF                MOV   T,#R0
164 0180 E3F1 E0E5            STTS #0101B         ;Clear Status Flag
165 0182 FFE0                MOV   A,R00

```

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 03-005

SOURCE = TRANS80.ASM

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
166 0183 F4F3             INC    A                ;Data Pointer L (R00) increment
167 0184 E5E0             MOV    R00,A
168 0185 E2F1 E0E5       HALT   #005H           ;HALT mode (Timer = 00H)
169 0187 E7FF             MOV    T,R0
170
171 0188 FFE7             MOV    A,R07
172 0189 F4F3             INC    A                ;Key Off Check Counter (R07) increment
173 018A ECF1 E9F5       JC     BIT_TRANS2      ;Key Off Check Counter (R07) = End of 10 Times ?
174                       ;      then BIT_TRANS2: (End of 10 Times)
175                       ;      else      (Less than 10 Times)
176 018C E3F1 E0EE       STTS   #1110B         ;With input key ?
177 018E EFF1 E9F4       JNF    BIT_TRANS1      ;      else BIT_TRANS1: (With input)
178                       ;** With input key ** ;      then      (Without input)
179 0190 E6E7 EFE5       MOV    R7,#0F5H       ;Key off Check Counter (R07) = 05H
180 0192 E8F1 E9F5       JMP    BIT_TRANS2
181                       ;
182                       ;** Without input key **
183 BIT_TRANS1:
184 0194 E5E7             MOV    R07,A
185
186 BIT_TRANS2:
187 0195 E3F1 E0E5       STTS   #0101B         ;Clear Status Flag
188 0197 FFEA             MOV    A,R0A
189 0198 F4F3             INC    A                ;Send Bit Counter (R0A) increment
190 0199 EDF1 E6F9       JNC    BITOUT
191 019B E8F2             RET
192                       ;
193 END

```

TOTAL ERRORS = 0
TOTAL WARNINGS = 0

END OF LIST

SOURCE = DTABLE80.TBL

```

E STNO LOC. OBJ.      M I SOURCE STATEMENT
1                      ;*****
2                      ;****          ****
3                      ;****      uPD6133 Series Data Table      ****
4                      ;****      Trans. CODE: NEC-R Format (80Key) ****
5                      ;****          ****
6                      ;*****
7      01FF          TIME9M      EQU      1FFH          ;9.00ms (9.002ms)
8
9                      ;#####      P U B L I C      #####
10                     PUBLIC      DATACODE
11                     PUBLIC      FS_27MS
12                     ;
13
14                     DATACODE:
15                     ;*****
16                     ;****      Data Code      ****
17                     ;****      Double Key      ****
18                     ;*****
19      019D          ORG      19DH
20                     ;** K29 + K30 **
21      019D E0E9          DW      009H          ;K29 + K30 : Key Data = 70H
22                     ;** K29 + K31 **
23      019E E8E9          DW      089H          ;K29 + K31 : Key Data = 71H
24                     ;** K29 + K32 **
25      019F E4E9          DW      049H          ;K29 + K32 : Key Data = 72H
26                     ;
27
28                     ;*****
29                     ;****      Frame Space      ****
30                     ;****      Time Data      ****
31      01A0 E7ED          ;ORG 1A0H *****
32                     DT      1F4H          ;Counter:000 = 8.815ms (8.809ms)
33      01A1 E6ED          DT      1B4H          ;Counter:001 = 7.690ms (7.683ms)
34      01A2 E5ED          DT      174H          ;Counter:010 = 6.565ms (6.558ms)
35      01A3 E4ED          DT      134H          ;Counter:011 = 5.440ms (5.433ms)
36      01A4 E3ED          DT      0F4H          ;Counter:100 = 4.315ms (4.308ms)
37      01A5 E2ED          DT      0B4H          ;Counter:101 = 3.190ms (3.182ms)
38      01A6 E1ED          DT      074H          ;Counter:110 = 2.065ms (2.057ms)
39      01A7 E0ED          DT      034H          ;Counter:111 = 0.940ms (0.932ms)
40                     ;
41
42                     ;*****
43                     ;****      Time Data      ****
44      01A8 *****
45                     ;** Leader Code **
46      01A8 FFFF          DT      3FFH          ;Carrier ON : 9.000ms (9.002ms)
47      01A9 F3FF          DT      0FFH          ;Carrier OFF : 4.500ms (4.501ms)
48                     ;** Bit data 0 **
49      01AA F8F7          DT      21FH          ;Carrier ON : 0.560ms (0.563ms)
50      01AB F0F7          DT      01FH          ;Carrier OFF : 0.565ms (0.563ms)
51                     ;** Bit data 1 **
52      01AC F8F7          DT      21FH          ;Carrier ON : 0.560ms (0.563ms)
53      01AD F1F7          DT      05FH          ;Carrier OFF : 1.690ms (1.687ms)
54                     ;** Stop Bit **
55      01AE F8F7          DT      21FH          ;Carrier ON : 0.560ms (0.563ms)

```

第9章 プログラム・リスト

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 04-003

SOURCE = DTABLE80.TBL

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
56 01AF E2FA              DT   0A9H          ;Carrier OFF : 3.000ms (2.989ms)
57                          ;
58
59
60                          ;*****
61                          ;****   Data Code   ****
62                          ;****   Single Key  ****
63                          ;*****
63 01B0                    ORG   1B0H
64                          ;** K33 - K40 **
65 01B0 E0E4              DW   004H          ;K33 : Data Code = 20H
66 01B1 E8E4              DW   084H          ;K34 : Data Code = 21H
67 01B2 E4E4              DW   044H          ;K35 : Data Code = 22H
68 01B3 ECE4              DW   0C4H          ;K36 : Data Code = 23H
69 01B4 E2E4              DW   024H          ;K37 : Data Code = 24H
70 01B5 EAE4              DW   0A4H          ;K38 : Data Code = 25H
71 01B6 E6E4              DW   064H          ;K39 : Data Code = 26H
72 01B7 EEE4              DW   0E4H          ;K40 : Data Code = 27H
73                          ;
74                          ;** K73 - K80 **
75 01B8 E1E2              DW   012H          ;K73 : Data Code = 48H
76 01B9 E9E2              DW   092H          ;K74 : Data Code = 49H
77 01BA E5E2              DW   052H          ;K75 : Data Code = 4AH
78 01BB EDE2              DW   0D2H          ;K76 : Data Code = 4BH
79 01BC E3E2              DW   032H          ;K77 : Data Code = 4CH
80 01BD EBE2              DW   0B2H          ;K78 : Data Code = 4DH
81 01BE E7E2              DW   072H          ;K79 : Data Code = 4EH
82 01BF EFE2              DW   0F2H          ;K80 : Data Code = 4FH
83                          ;
84
85 01C0                    ORG   1C0H
86                          ;** K 1 - K16 **
87 01C0 E0E0              DW   000H          ;K 1 : Data Code = 00H
88 01C1 E8E0              DW   080H          ;K 2 : Data Code = 01H
89 01C2 E4E0              DW   040H          ;K 3 : Data Code = 02H
90 01C3 ECE0              DW   0C0H          ;K 4 : Data Code = 03H
91 01C4 E2E0              DW   020H          ;K 5 : Data Code = 04H
92 01C5 EAE0              DW   0A0H          ;K 6 : Data Code = 05H
93 01C6 E6E0              DW   060H          ;K 7 : Data Code = 06H
94 01C7 EEE0              DW   0E0H          ;K 8 : Data Code = 07H
95 01C8 E1E0              DW   010H          ;K 9 : Data Code = 08H
96 01C9 E9E0              DW   090H          ;K10 : Data Code = 09H
97 01CA E5E0              DW   050H          ;K11 : Data Code = 0AH
98 01CB EDE0              DW   0D0H          ;K12 : Data Code = 0BH
99 01CC E3E0              DW   030H          ;K13 : Data Code = 0CH
100 01CD EBE0             DW   0B0H          ;K14 : Data Code = 0DH
101 01CE E7E0             DW   070H          ;K15 : Data Code = 0EH
102 01CF EFE0             DW   0F0H          ;K16 : Data Code = 0FH
103                          ;
104                          ;** K17 - K32 **
105 01D0 E0E8             DW   008H          ;K17 : Data Code = 10H
106 01D1 E8E8             DW   088H          ;K18 : Data Code = 11H
107 01D2 E4E8             DW   048H          ;K19 : Data Code = 12H
108 01D3 ECE8             DW   0C8H          ;K20 : Data Code = 13H
109 01D4 E2E8             DW   028H          ;K21 : Data Code = 14H
110 01D5 EAE8             DW   0A8H          ;K22 : Data Code = 15H

```

SOURCE = DTABLE80.TBL

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
111 01D6 E6E8             DW      068H          ;K23 : Data Code = 16H
112 01D7 EEE8             DW      0E8H          ;K24 : Data Code = 17H
113 01D8 E1E8             DW      018H          ;K25 : Data Code = 18H
114 01D9 E9E8             DW      098H          ;K26 : Data Code = 19H
115 01DA E5E8             DW      058H          ;K27 : Data Code = 1AH
116 01DB EDE8             DW      0D8H          ;K28 : Data Code = 1BH
117 01DC E3E8             DW      038H          ;K29 : Data Code = 1CH
118 01DD EBE8             DW      0B8H          ;K30 : Data Code = 1DH
119 01DE E7E8             DW      078H          ;K31 : Data Code = 1EH
120 01DF EFE8             DW      0F8H          ;K32 : Data Code = 1FH
121                          ;
122                          ;** K41 - K56 **
123 01E0 E1E4             DW      014H          ;K41 : Data Code = 28H
124 01E1 E9E4             DW      094H          ;K42 : Data Code = 29H
125 01E2 E5E4             DW      054H          ;K43 : Data Code = 2AH
126 01E3 EDE4             DW      0D4H          ;K44 : Data Code = 2BH
127 01E4 E3E4             DW      034H          ;K45 : Data Code = 2CH
128 01E5 EBE4             DW      0B4H          ;K46 : Data Code = 2DH
129 01E6 E7E4             DW      074H          ;K47 : Data Code = 2EH
130 01E7 EFE4             DW      0F4H          ;K48 : Data Code = 2FH
131 01E8 E0EC             DW      00CH          ;K49 : Data Code = 30H
132 01E9 E8EC             DW      08CH          ;K50 : Data Code = 31H
133 01EA E4EC             DW      04CH          ;K51 : Data Code = 32H
134 01EB ECEC             DW      0CCH          ;K52 : Data Code = 33H
135 01EC E2EC             DW      02CH          ;K53 : Data Code = 34H
136 01ED EAEC             DW      0ACH          ;K54 : Data Code = 35H
137 01EE E6EC             DW      06CH          ;K55 : Data Code = 36H
138 01EF EECC             DW      0ECC          ;K56 : Data Code = 37H
139                          ;
140                          ;** K57 - K72 **
141 01F0 E1EC             DW      01CH          ;K57 : Data Code = 38H
142 01F1 E9EC             DW      09CH          ;K58 : Data Code = 39H
143 01F2 E5EC             DW      05CH          ;K59 : Data Code = 3AH
144 01F3 EDEC             DW      0DCH          ;K60 : Data Code = 3BH
145 01F4 E3EC             DW      03CH          ;K61 : Data Code = 3CH
146 01F5 EBEC             DW      0BCH          ;K62 : Data Code = 3DH
147 01F6 E7EC             DW      07CH          ;K63 : Data Code = 3EH
148 01F7 EFEC             DW      0FCH          ;K64 : Data Code = 3FH
149 01F8 E0E2             DW      002H          ;K65 : Data Code = 40H
150 01F9 E8E2             DW      082H          ;K66 : Data Code = 41H
151 01FA E4E2             DW      042H          ;K67 : Data Code = 42H
152 01FB ECE2             DW      0C2H          ;K68 : Data Code = 43H
153 01FC E2E2             DW      022H          ;K69 : Data Code = 44H
154 01FD EAE2             DW      0A2H          ;K70 : Data Code = 45H
155 01FE E6E2             DW      062H          ;K71 : Data Code = 46H
156 01FF EEE2             DW      0E2H          ;K72 : Data Code = 47H
157                          ;
158
159
160
161
162                          ;*****
163                          ;*****      Frame Space = 27ms      *****
164                          ;*****      Subroutine : FS 27ms      *****
165                          ;*****

```

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 04-005

SOURCE = DTABLE80.TBL

```

E STNO LOC. OBJ.          M I SOURCE STATEMENT
166                      ;*****
167                      FS_27MS:
168 0200 FFF1 FDE0          MOV     A,#0DH          ;Counter(Acc) = 3 Times (=0DH)
169                      FS_27MS0:
170 0202 E2F1 E0E5          HALT   #005H          ;HALT mode (Timer = 00H)
171 0204 E6FF F7FF          MOV     T,#TIME9M     ;Timer = 9.00ms
172 0206 E3F1 E0E5          STTS   #0101B        ;Status Flag Clear
173 0208 F4F3              INC     A              ;More than 27ms ?
174 0209 EDF1 F0E2          JNC    FS_27MS0      ;     else FS_27MS0: (Frame Space < 27ms)
175 020B E8F2              RET                     ;     then RET      (Frame Space = 27ms)
176                      ;
177                      END
    
```

TOTAL ERRORS = 0
 TOTAL WARNINGS = 0

END OF LIST

AS6133 V1.01 << D6134 ASSEMBLE LIST >> 01:00:01 08/29/96 PAGE 05-002

SOURCE = OPTION.ASM

E	STNO	LOC.	OBJ.	M	I	SOURCE STATEMENT
	1					*****
	2					***** Set Option *****
	3					*****
	4					OPTION
	5	0000	0001			USEPOC
	6					ENDOP
	7					END

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

アンケート記入のお願い

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

[ドキュメント名] μPD6133シリーズ リモコン送信プログラム例 アプリケーション・ノート
(U11726JJ1V0AN00 (第1版))

[お名前など] (さしつかえない範囲で)

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 好	普 通	悪 劣	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他()					
()					

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは

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

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡しください。

NEC半導体インフォメーションセンター

FAX: (044) 548-7900

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

半導体第一販売事業部	〒108-01	東京都港区芝五丁目7番1号 (NEC本社ビル)	東京	(03)3454-1111	(大代表)
半導体第二販売事業部					
半導体第三販売事業部					
中部支社 半導体販売部	〒460	名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋	(052)222-2170	
関西支社 半導体第一販売部	〒540	大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪	(06) 945-3178	
半導体第二販売部			大阪	(06) 945-3200	
半導体第三販売部			大阪	(06) 945-3208	
北海道支社 札幌	(011)231-0161	宇都宮支店 宇都宮	(028)621-2281	富山支店 富山	(0764)31-8461
東北支社 仙台	(022)261-5511	小山支店 小山	(0285)24-5011	三重支店 津	(0592)25-7341
岩手支店 盛岡	(0196)51-4344	長野支社 長野	(026)235-1444	京都支社 京都	(075)344-7824
山形支店 山形	(0236)23-5511	松本支店 松本	(0263)35-1666	神戸支社 神戸	(078)333-3854
郡山支店 郡山	(0249)23-5511	上諏訪支店 上諏訪	(0266)53-5350	中国支社 広島	(082)242-5504
いわき支店 いわき	(0246)21-5511	甲府支店 甲府	(0552)24-4141	鳥取支店 鳥取	(0857)27-5311
長岡支店 長岡	(0258)36-2155	埼玉支社 埼玉	(048)641-1411	岡山支店 岡山	(086)225-4455
土浦支店 土浦	(0298)23-6161	立川支社 立川	(0425)26-5981	四国支社 高松	(0878)36-1200
水戸支店 水戸	(0292)26-1717	千葉支社 千葉	(043)238-8116	新居浜支店 新居浜	(0897)32-5001
神奈川支社 横浜	(045)324-5511	静岡支社 静岡	(054)255-2211	松山支店 松山	(089)945-4111
群馬支店 高崎	(0273)26-1255	北陸支社 北陸	(0762)23-1621	九州支社 福岡	(092)271-7700
太田支店 太田	(0276)46-4011	福井支店 福井	(0776)22-1866	北九州支店 北九州	(093)541-2887

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部	〒210	川崎市幸区塚越三丁目484番地	川崎	(044)548-7923	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお願い致します)
マイクロコンピュータ技術部					
半導体販売技術本部	〒108-01	東京都港区芝五丁目7番1号 (NEC本社ビル)	東京	(03)3798-9619	
東日本販売技術部					
半導体販売技術本部	〒460	名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋	(052)222-2125	
中部販売技術部					
半導体販売技術本部	〒540	大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪	(06) 945-3383	
西日本販売技術部					