

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、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。

H8/300L SLP シリーズ

赤外線リモコン

要旨

赤外光トランシーバを接続したマイコンと、パソコンの IrDA 通信ポートを使い、物理層による文字の送受信を行う。

動作確認デバイス

H8/38024

目次

1. 仕様	2
2. 考え方	3
3. 使用機能説明	4
4. 動作説明	13
5. ソフトウェア説明	15
6. フローチャート	21
7. プログラムリスト	30

1. 仕様

- (1) タイマ C, タイマ F を用いて, 赤外線リモコン送信を行います。
- (2) タイマ G を用いて, 赤外線リモコン受信を行います。
- (3) 本タスク例では, 送信側に接続したキースイッチ(0~9,A~F)を押し, そのキー番号を受信側 LCD に表示します。
- (4) 赤外線リモコンは, 半二重通信です。
- (5) 赤外線リモコンの接続図を図 1 に示します。

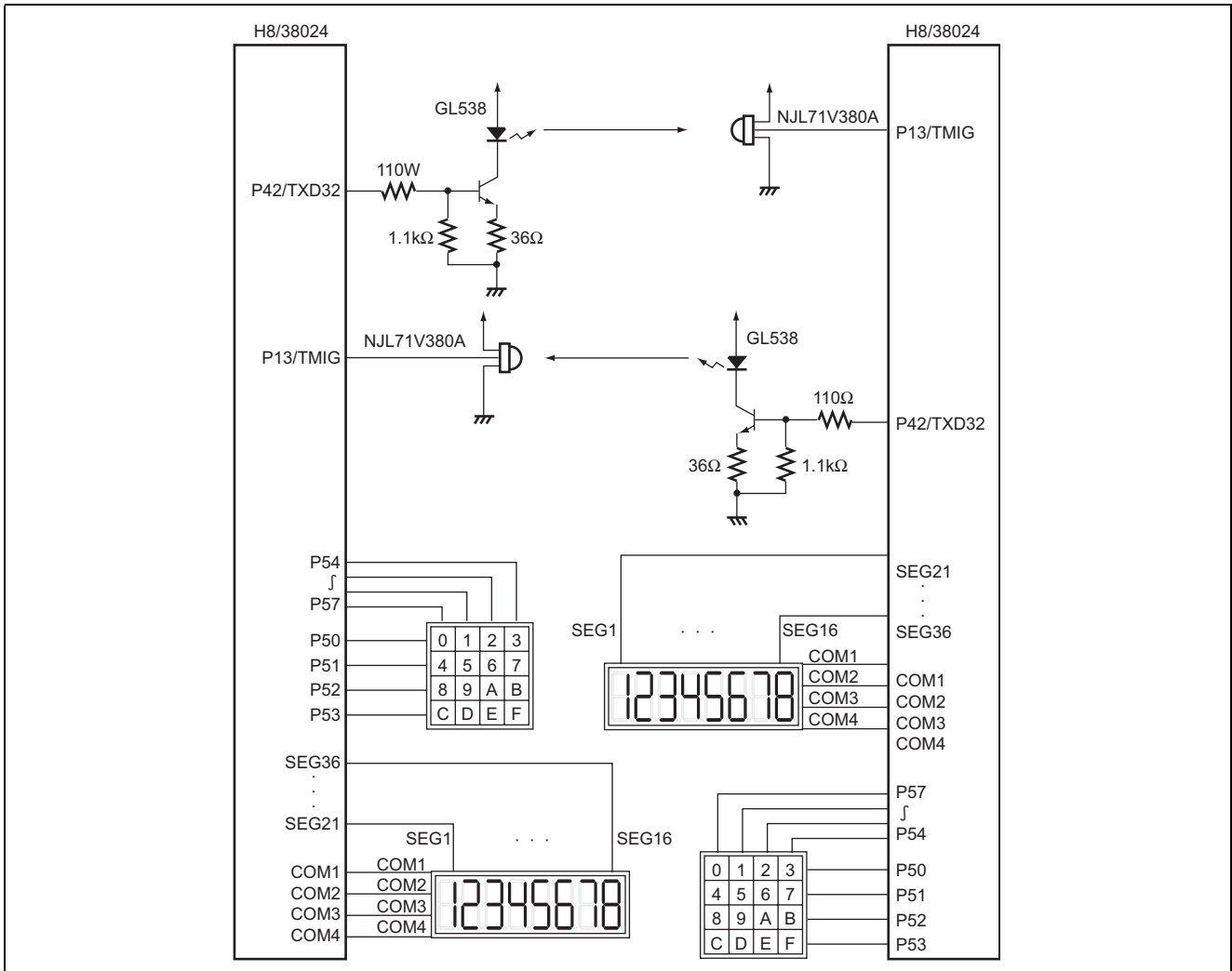


図 1 赤外線リモコンの接続例

2. 考え方

(1) 本タスク例で使用する赤外線リモコン処理の信号例を図2に示します。本タスク例で使用する赤外線受光素子は、対応する周波数(搬送波 38kHz)を受信したとき、Low レベルを出力します。

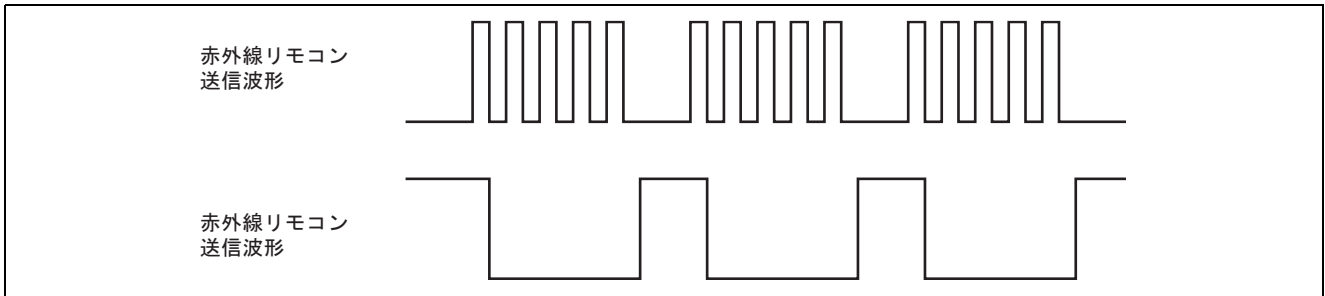


図2 赤外線リモコン処理の信号例

(2) 0と1の判定

本タスク例では、搬送波の長さによって0と1を判別します。

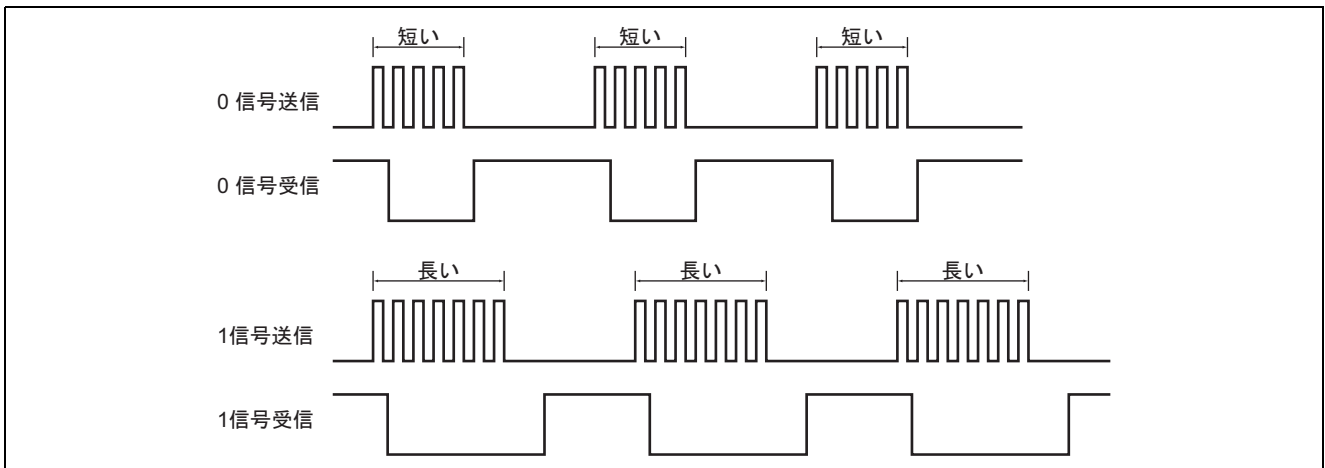


図3 0と1の判定方法

3. 使用機能説明

- (1) 本タスク例で使用する赤外発光ダイオード GL538 の仕様を表 1 に示します。また、本タスク例で使用する赤外線受光素子 NJL71V380A の仕様を表 2 に示します。(If:順電流, Ifm:ピーク順電流, Vr 逆電圧)

表 1 赤外発光ダイオードの仕様

項目	条件	値,名称
型名	—	GL538
メーカー	—	SHARP(株)
動作電源電圧 Vf	If=50mA	1.3V ~ 1.5V
ピーク順電圧 Vfm	Ifm=0.5A	1.9 ~ 3.0V
逆電流 Ir	Vr=3V	10 μ A
ピーク発光波長 λ_p	If=5mA	950nm
半値波長	If=5mA	45nm
発光出力 Ie	If=50mA	15 ~ 30mW/sr
半値角	If=20mA	± 13 deg
発振特性	—	300kHz

表 2 赤外線受光素子の仕様

項目	条件	値,名称
型名	—	NJL71V380A
メーカー	—	新日本無線(株)
搬送周波数	—	38kHz
動作電源電圧	—	2.4V ~ 5.5V
消費電流	入射光なし	0.6mA(最大値)
到達距離	光軸中心方向, 搬送周波数	18m
半値角(上下)	到達距離の 1/2 となる左右指向性	45deg
	到達距離の 1/2 となる上下指向性	30deg
出力ローレベル電圧	光軸方向 30cm	0.5V(最大値)
出力ハイレベル電圧	光軸方向 30cm	2.8V
ローレベルパルス幅(TwL)	光軸方向 30cm から到達距離までの範囲での出力 TwL, TwH の期間幅で規定する。 (50 パルスの平均値)	350 ~ 800 μ s
ハイレベルパルス幅(TwH)		400 ~ 850 μ s

(2) 赤外線リモコンの機能について説明します。

(a) 赤外線リモコン処理の H8/38024 使用機能ブロック図を図 4 に示します。

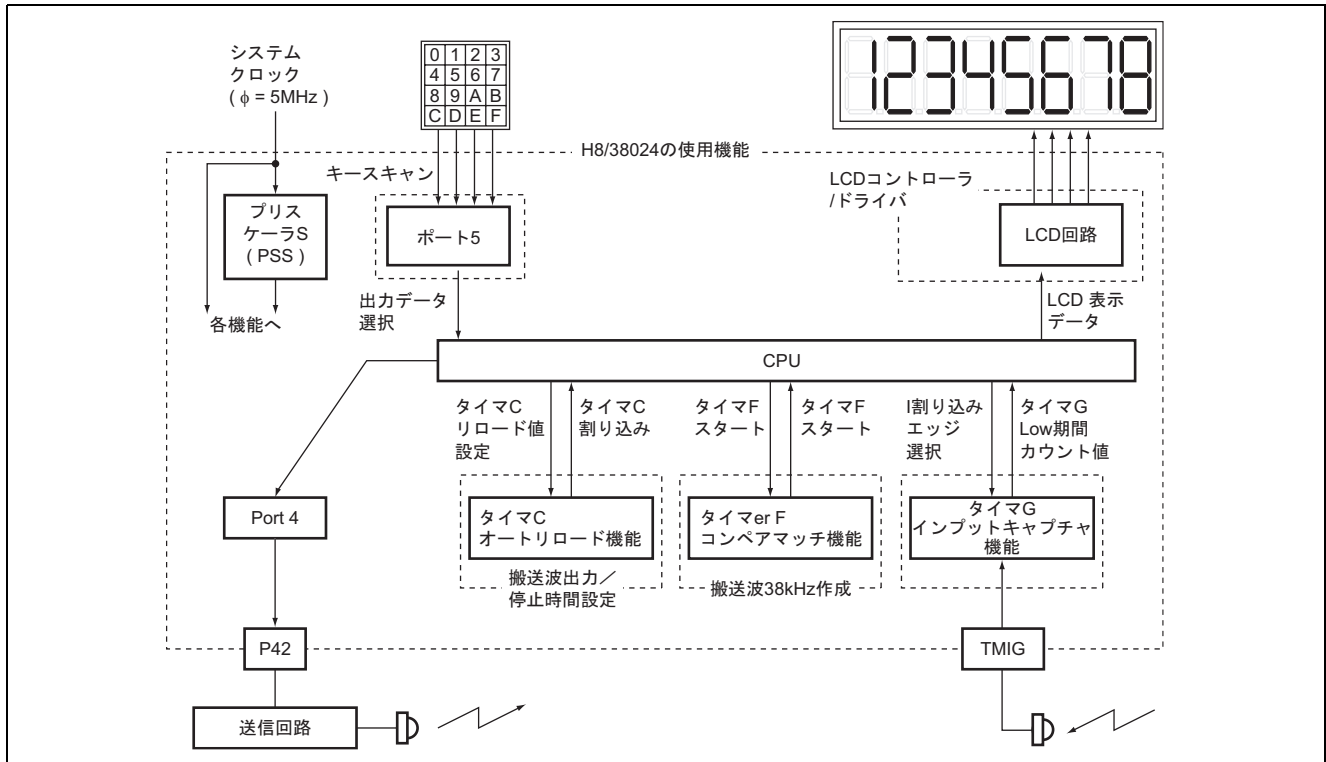


図 4 赤外線リモコン送信処理のブロック図

(b) 以下に赤外線リモコンで使用する H8/38024 の機能について説明します。

- システムクロック ()
10MHz の OSC クロックを 2 分周した 5MHz のクロックで、CPU および周辺機能を動作させるための基準クロックです。
- プリスケラ S(PSS)
を入力とする 13 ビットのカウンタで、1 サイクルごとにカウントアップします。
- ポート 4
P42 を出力端子とし、搬送波を出力します。
- ポート 5
キースイッチと接続しており、赤外線送信データを選択します。
- タイマ C オートリロード機能
搬送波の出力/停止間隔を制御します。
- タイマ F コンペアマッチ機能
38kHz の搬送波を作成します。
- タイマ G インプットキャプチャ機能
赤外線受光素子の出力値を入力し、受信データの判定を行います。
- LCD コントローラ/ドライバ
受信データを LCD に表示します。

(3) ポート 4 の機能を説明します。

(a) ポート 42 による搬送波送信のブロック図をに示します。

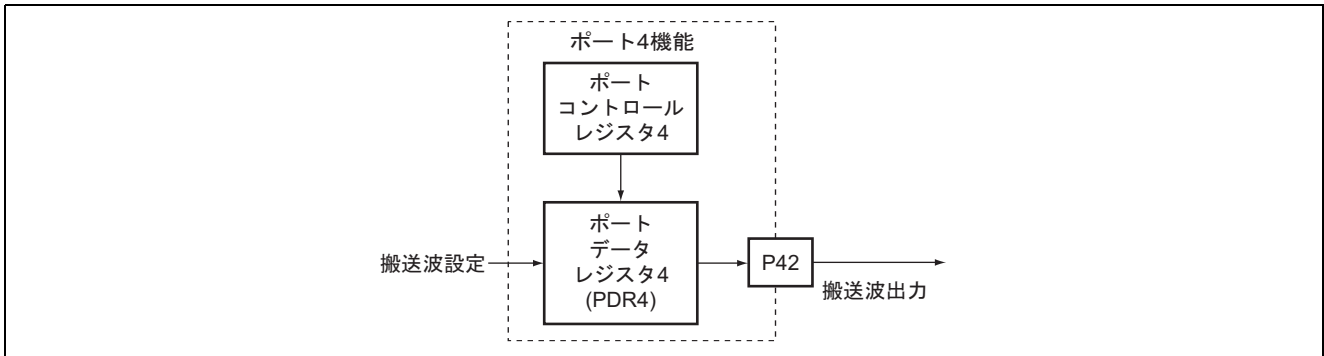


図 5 ポート 4 のブロック図

(b) 本タスク例では、ポート 42 を使用して、搬送波を出力します。以下にポート 4 のブロック図について説明します。

- ポートコントロールレジスタ 4(PCR4)
ポート 4 の入出力設定を行います。PCR4=0xFC とし、P42 を出力ポートに設定します。
- ポートデータレジスタ 4(PDR4)
出力ポート P42 に格納するデータを設定します。タイマ F 割り込みにより P42 をトグルに切り換え、38kHz の搬送波を出力します。

(4) ポート 5 の機能を説明します。

(a) ポート 5 によるキー入力回路のブロック図を図 6 に示します。

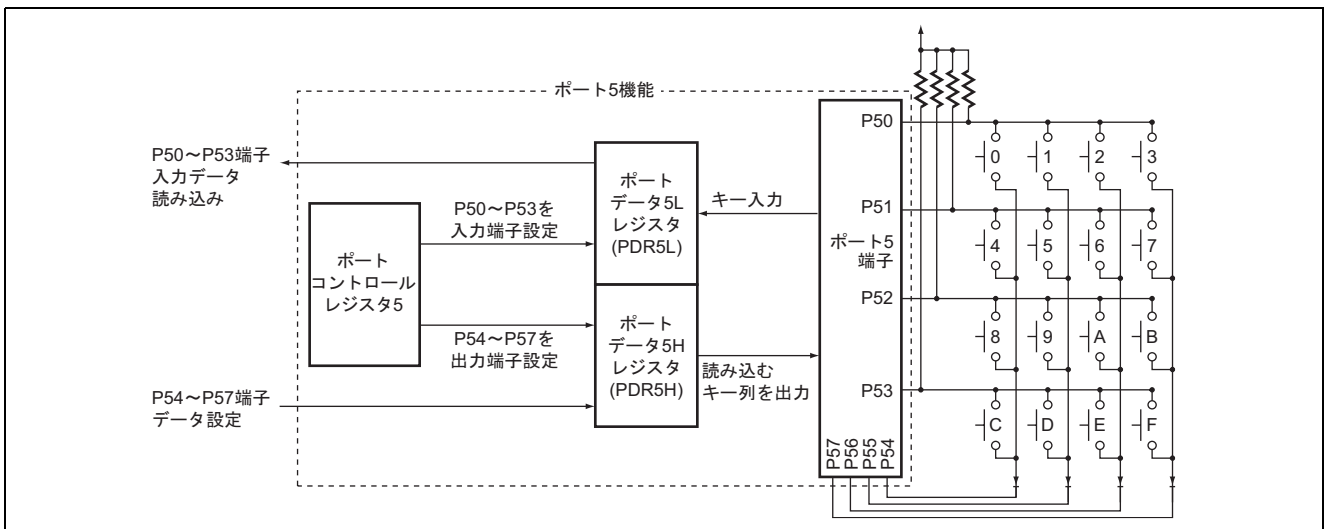


図 6 ポート 5 によるキー入力回路のブロック図

(b) 本タスク例では、ポート 5 を使用して、キー入力を行います。以下にキー入力回路のブロック図について説明します。

- ポートコントロールレジスタ 5(PCR5)
ポート 5 の入出力設定を行います。PCR5=0xF0 とし、P57 ~ P54 を出力ポート、P53 ~ P50 を入力ポートに設定します。
- ポートデータレジスタ 5H(PDR5H)
PDR5 の上位 4 ビットを PDR5H とし、出力ポート P57 ~ P54 に格納するデータを設定します。P57~P54 のうち、Low レベルに設定したキー列の状態が P53 ~ P50 に反映されます。
- ポートデータレジスタ 5L(PDR5L)
PDR5 の下位 4 ビットを PDR5L とし、入力ポート P53 ~ P50 の値は、PDR5L に反映されます。PDR5L に格納された値は、PDR5H で選択されたキー列の状態です。

(5) タイマ C オートリロード機能を説明します。

(a) タイマ C オートリロード機能のブロック図を図 7 に示します。

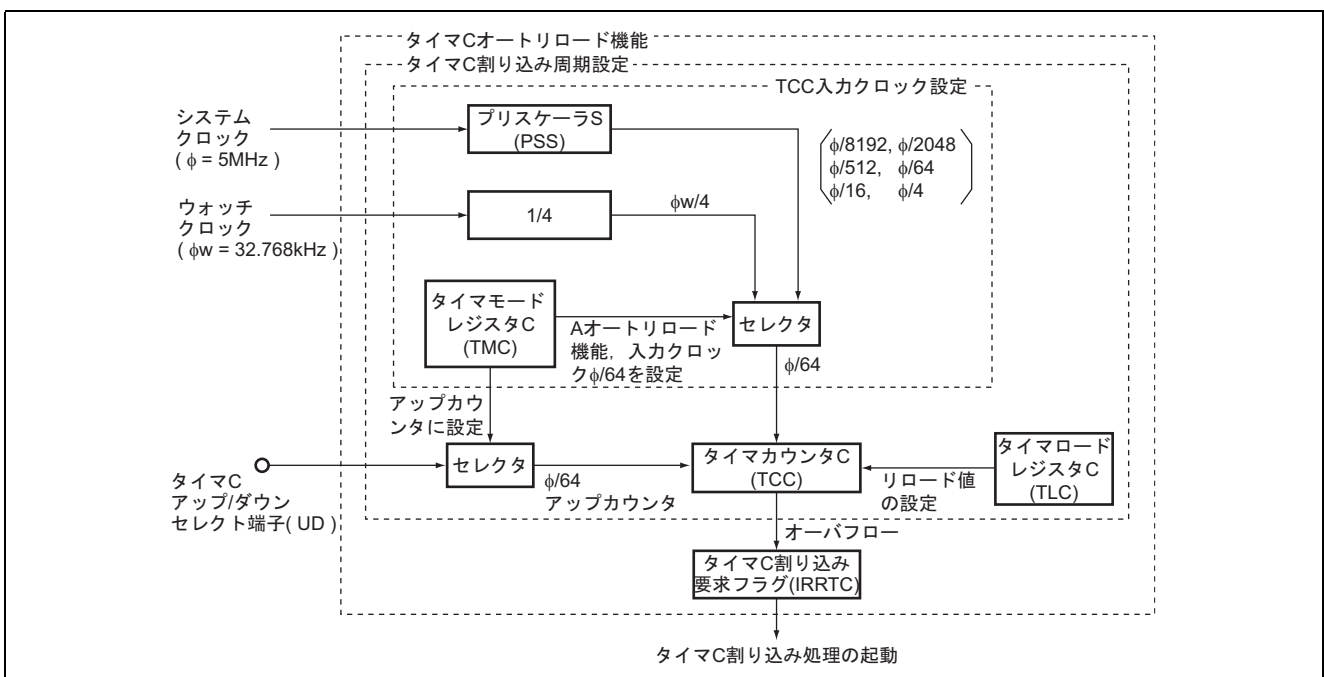


図 7 タイマ C オートリロード機能のブロック図

(b) 以下にタイマ C オートリロード機能のブロック図について説明します。

- タイマモードレジスタ C(TMC)
8ビットのリード/ライト可能なレジスタで、オートリロード機能の選択、タイマカウンタ C(TCC)のアップ/ダウン制御、および入力クロックの選択を行います。TCCのアップ/ダウン制御は、UD 端子入力によるハードウェア制御とするか、ソフトウェア制御によるアップカウンタまたはダウンカウンタとするかを選択します。
- タイマカウンタ C(TCC)
8ビットのリード可能なカウンタで、入力する内部クロック/外部イベントにより、カウントアップ/ダウンされます。入力するクロックはシステムクロックの 8192, 2048, 512, 64, 16, 4 分周、ウォッチクロックの 4 分周、および外部クロックの計 8 種類から選択可能です。本タスク例では、TCC をアップカウンタに設定し、TCC の入力クロックにシステムクロックの 64 分周を選択しています。
- タイマロードレジスタ C(TLC)
8ビットのライト専用のレジスタで、TCC のリロード値を設定します。
- タイマ C 割り込み要求フラグ(IRRTC)
TCC がオーバフローすることにより"1"にセットされます。IRRTC が"1"にセットされていて、割り込み許可レジスタ 2(IENR2)のタイマ C 割り込みイネーブル(IENTC)が"1"で、かつコンディショニングコードレジスタ(CCR)の I ビットが"0"にクリアされている場合にタイマ C 割り込みが受け付けられ、タイマ C 割り込み処理を開始します。

(6) タイマ F アウトプットコンペア機能を説明します。

(a) タイマ F アウトプットコンペア機能のブロック図を図 8 に示します。

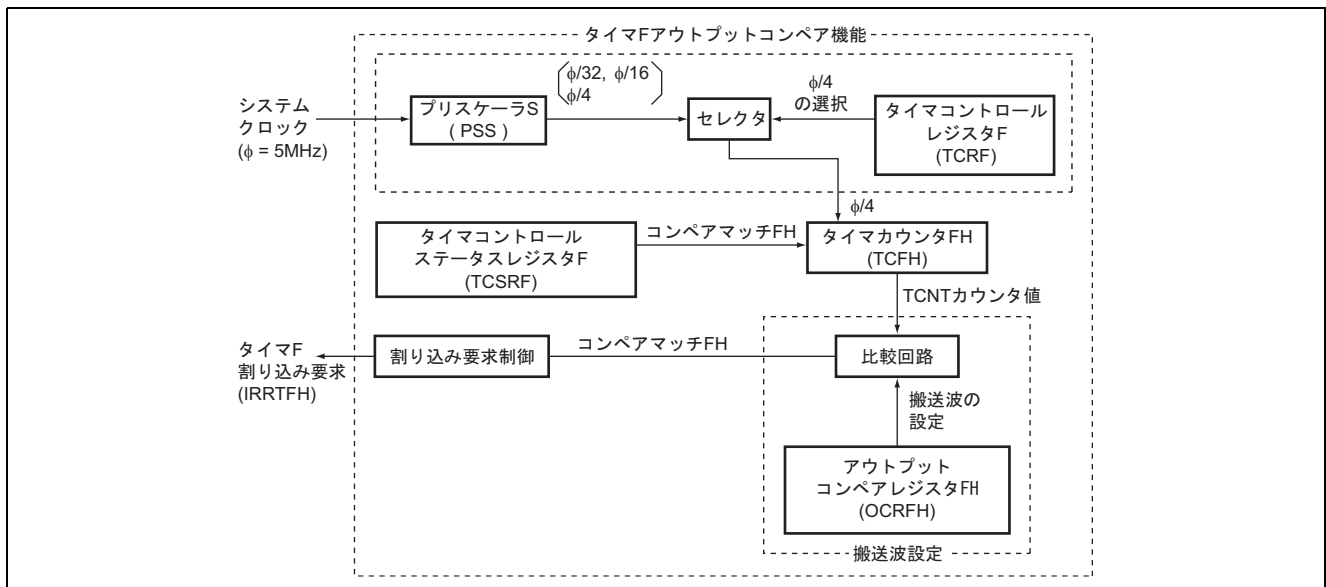


図 8 タイマ F アウトプットコンペア機能のブロック図

(b) 以下にタイマ F アウトプットコンペア機能のブロック図について説明します。

- タイマカウンタ FH(TCFH)
8ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは、 ϕ を分周した 4 種類の内部クロックと、1 つの外部クロックを選択できます。
- タイマコントロールレジスタ F(TCRF)
8ビットのライト可能なレジスタで、TCFH の入力クロックの選択を設定します。本タスク例では、TCFH の入力クロックに ϕ の 4 分周を選択しています。
- タイマコントロールステータスレジスタ F(TCSR F)
8ビットのレジスタで、カウンタクリアの選択、コンペアマッチフラグのセット、タイマオーバフローフラグのセット、およびオーバフローによる割り込み要求許可の制御を行います。本タスク例では、コンペアマッチ FH による TCFH のクリアを許可に、タイマ FH オーバフローによる割り込みを不許可にしています。
- アウトプットコンペアレジスタ FH(OCR FH)
8ビットのリード/ライト可能なレジスタで、OCR FH の内容は TCFH と常に比較されています。両者の値が一致すると、TCSR F のコンペアマッチフラグ H(CMFH)が"1"にセットされます。このとき、コンペアマッチ FH が発生し、CPU に割り込みを要求します。

(7) タイマ G インプットキャプチャ機能を説明します。

(a) タイマ G インプットキャプチャ機能のブロック図を図 9 に示します。

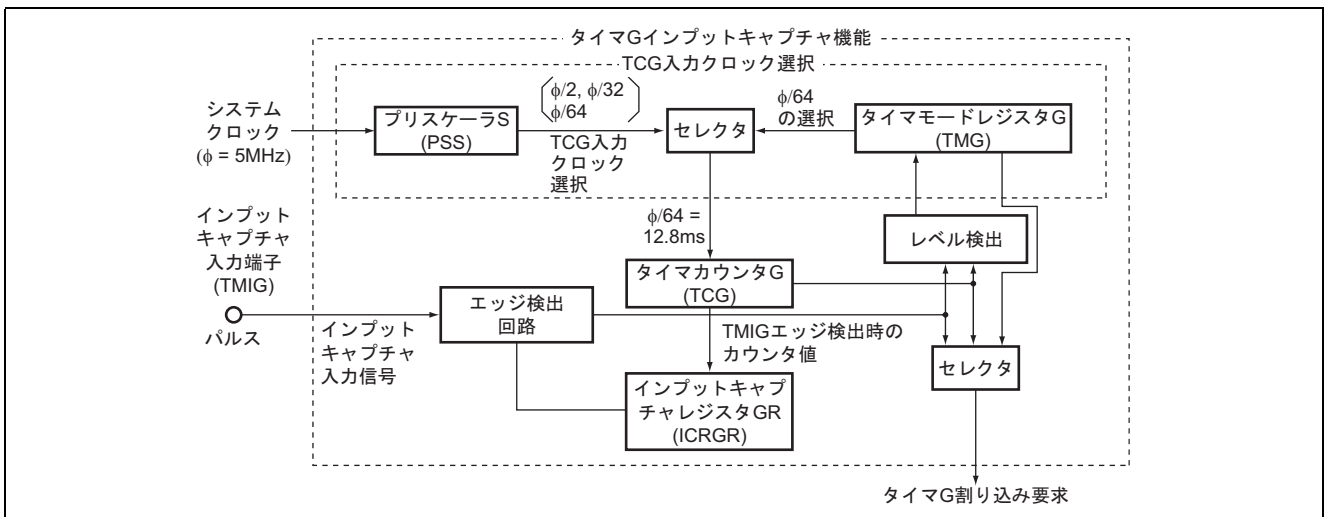


図 9 タイマ G インプットキャプチャ機能のブロック図

(b) タイマ G インputキャプチャ機能について説明します。

- タイマカウンタ G(TCG)

8ビットのリード/ライト不可なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは、システムクロックの2分周、32分周、64分周および外部クロックの計4種類のクロックを選択可能です。本タスク例では、TCGの入力クロックにシステムクロックの64分周を選択しています。
- タイマモードレジスタ G(TMG)

8ビットのリード/ライト可能なレジスタで、TCGの入力クロックの選択、カウンタクリアの選択、インプットキャプチャ入力信号による割り込み要求のエッジ選択、オーバフローによる割り込み要求の許可/禁止を制御、およびオーバフローフラグの表示を行います。
- インプットキャプチャレジスタ GR(ICRGR)

8ビットのリード専用のレジスタで、インプットキャプチャ入力信号の立ち上がりエッジが検出されると、そのときのTCGの値がICRGRに転送されます。このとき、IRR2のIRRTGが"1"ならばCPUに割り込みを要求します。
- インプットキャプチャ入力端子(TMIG)

赤外線受光素子の出力値を入力し、Low期間、High期間を測定します。

(8) LCDコントローラ/ドライバについて説明します。

(a) LCDコントローラ/ドライバのブロック図を図10に示します。

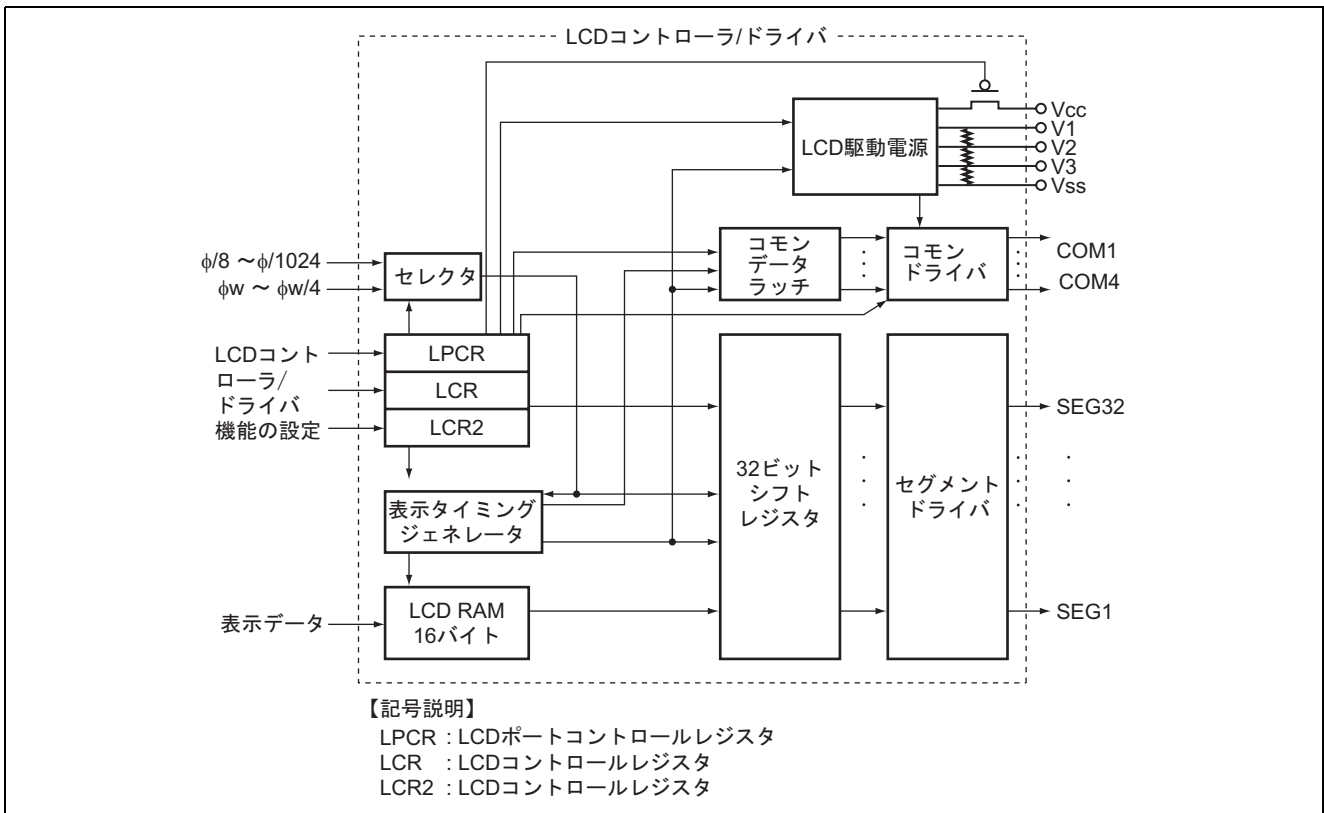


図10 LCDコントローラ/ドライバのブロック図

(b) LCD コントローラ/ドライバの各機能について説明します。

- LCD ポートコントロールレジスタ(LPCR)
- LPCR は、8 ビットのリード/ライト可能なレジスタです。デューティ比の選択、LCD ドライバや端子機能の選択を行います。
- LCD コントロールレジスタ(LCR)
- LCR は、8 ビットのリード/ライト可能なレジスタです。LCD 駆動電源の ON/OFF 制御、表示機能開始制御、表示データの制御、フレーム周波数選択を行います。
- LCD コントロールレジスタ 2(LCR2)
- LCR2 は、8 ビットのリード/ライト可能なレジスタです。A 波形/B 波形切り換えの制御、3 倍昇圧回路のクロック選択、駆動電源の選択、電源分割抵抗を電源回路に接続している期間のデューティ比選択を行います。
- セグメント出力端子(SEG32 ~ SEG1)
- LCD のセグメント駆動用の端子です。全端子、ポートと兼用でプログラマブルに設定可能です。
- コモン出力端子(COM4 ~ COM1)
- LCD のコモン駆動端子です。スタティック、1/2 デューティ時には端子の並列化が可能です。
- LCD 電源端子(V1, V2, V3)
- 外付けでパソコンを接続する場合、外部電源回路を使用する場合に使用します。
- LCDRAM
- 表示データを設定します。また、LCDRAM と表示セグメントの関係は、デューティ比によって異なります。表示に必要なレジスタ群を設定した後、デューティに対応する部分に通常の RAM と同様な命令によってデータを書き込み、表示を ON すれば自動的に表示を開始します。RAM 設定にはワード/バイトアクセス命令が使用できます。

(c) 3 桁 8 セグメント LCD の SEG21, SEG22 の表示と表示データ例を表 3 に示します。

表 3 表示データ例

記号	表示	番地	表示データ								
			2 進数							16 進数	
0	0	0xF746	1	1	0	1	0	1	1	1	0xD7
1	1	0xF746	0	0	0	0	0	1	1	0	0x06
2	2	0xF746	1	1	1	0	0	0	1	1	0xE3
3	3	0xF746	1	0	1	0	0	1	1	1	0xA7
4	4	0xF746	0	0	1	1	0	1	1	0	0x36
5	5	0xF746	1	0	1	1	0	1	0	1	0xB5
6	6	0xF746	1	1	1	1	0	1	0	1	0xF5
7	7	0xF746	0	0	0	0	0	1	1	1	0x07
8	8	0xF746	1	1	1	1	0	1	1	1	0xF7
9	9	0xF746	1	0	1	1	0	1	1	1	0xB7
A	A	0xF746	0	1	1	1	0	1	1	1	0x77
B	b	0xF746	1	1	1	1	0	1	0	0	0xF4
C	c	0xF746	1	1	0	1	0	0	0	1	0xD1
D	d	0xF746	1	1	1	0	0	1	1	0	0xE6
E	e	0xF746	1	1	1	1	0	0	0	1	0xF1
F	F	0xF746	0	1	1	1	0	0	0	1	0x71

(9) 本タスク例の機能割り付けを表 4 に示します。表 4 に示すように機能を割り付け，赤外線リモコン処理を行います。

表 4 送信側-機能割り付け

機能	機能割り付け
PSS	システムクロックを入力とする 13 ビットのカウンタ
TMC	タイマ C オートリロードレジスタ機能設定，入力クロック設定
TCC	タイマ C のカウンタ
TLC	搬送波の出力/停止時間を設定
TCRF	タイマ FH の入力クロック設定
TCSRFB	コンペアマッチによる TCFH クリアを設定
TCFH	タイマ FH のカウンタ
OCRFB	搬送波のタイミングを作成
PCR4	P42 を出力ポートに設定
PDR4(P42)	搬送波出力端子
PMR5	ポート 5 の端子機能切り換え
PDR5	キースイッチの入力端子
PUCR5	ポート 5 のプルアップ MOS を OFF にする
PCR5	ポート 5 の入出力設定
TMG	TCG のクリア，入力クロック，割り込みエッジを設定
TCG	タイマ G のカウンタ
ICRGR	赤外線受光素子の受信データを格納
LPCR	LCD のデューティ比，セグメント端子を選択
LCR	LCD の ON/OFF，フレーム周波数を設定
LCR2	LCD の A/B 波形を選択
LCDRAM	LCD の表示データを格納
TMIG	赤外受光素子の出力値を入力する端子
NCS	インプットキャプチャ入力信号のノイズ除去機能を ON/OFF 設定
IENFG	インプットキャプチャ入力信号による割り込み要求の許可
IENFB	タイマ FH コンペアマッチによる割り込み要求の許可
IENFC	タイマ C オーバフローによる割り込み要求の許可
IRRTG	インプットキャプチャ入力信号による割り込みフラグ
IRRTFB	タイマ FH コンペアマッチによる割り込みフラグ
IRRTC	タイマ C オーバフローによる割り込みフラグ

4. 動作説明

(1) 赤外線リモコンのフローチャートを図 11 に示します。図 11 に示す手順で赤外線リモコン処理を行います。

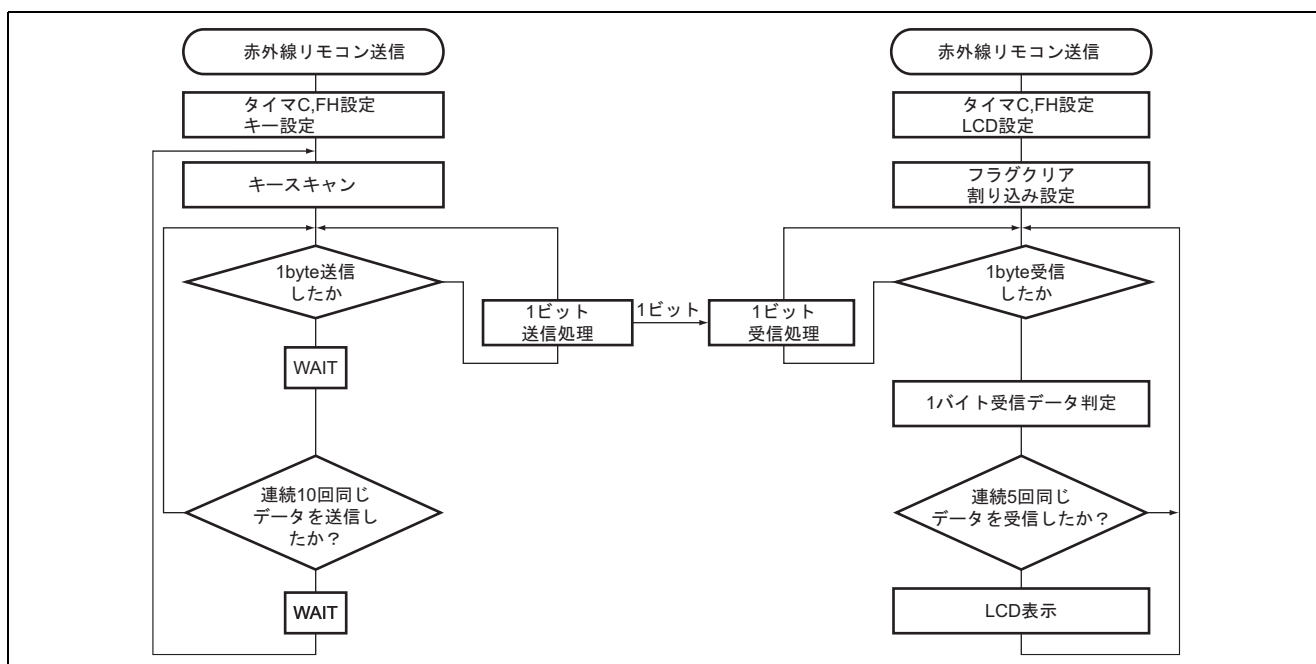


図 11 赤外線リモコンのフローチャート

(2) 0/1 判定方法

搬送波の長さで 0 と 1 を判別します。

送信ビット	搬送波出力時間
0	380 μs
1	770 μs

また、1 ビット送信間の搬送波停止時間は、600 μs で一定です。

- タイマ C の設定
 - タイマ C の入力クロック： $1/64 = 78.1\text{kHz}$
 - タイマ C の 1 カウント時間： $1/78.1\text{kHz} = 12.8\ \mu\text{s}$
- 送信ビット"0"(380 μs)の設定方法
 - 380 μs に必要なタイマ C カウント数
 - $380\ \mu\text{s} / 12.8\ \mu\text{s} = 29.6 = 30$
 - TLC の設定
 - (アップカウンタのため) $256 - 30 = 226$
- 送信ビット"1"(770 μs)の設定方法
 - 770 μs に必要なタイマ C カウント数
 - $770\ \mu\text{s} / 12.8\ \mu\text{s} = 60.1 = 61$
 - TLC の設定
 - (アップカウンタのため) $256 - 61 = 195$
- 1 ビット送信間の搬送波停止時間(600 μs)の設定方法
 - 600 μs に必要なタイマ C カウント数
 - $600\ \mu\text{s} / 12.8\ \mu\text{s} = 46.8 = 47$
 - TLC の設定
 - (アップカウンタのため) $256 - 47 = 209$

(3) 1 バイト確定方法

送信側から同じデータを 10 回送信し、そのうち連続 5 回受信したとき、正しい値として、データを格納します。

(4) 1 バイト送信間隔

1 バイト送信間の搬送波停止時間は、2ms とします。

受信側では、搬送波停止時間が 2ms 以上となった場合、データの先頭ビットと判断します。

- 1 バイト送信間の搬送波停止時間(2ms)の設定方法

2ms に必要なタイマ C カウント数

$$2000 \mu s / 12.8 \mu s = 156.2 = 157$$

TLC の設定

(アップカウンタのため) $256 - 157 = 99$

(5) 受信側の再送/新規データ判定

- 搬送波が 3.2ms 以上停止した場合、新規データと判断します。

5. ソフトウェア説明

(1) モジュール説明

本タスク例のモジュールを表 5 に示します。

表 5 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	割り込み設定, LCD 初期化, 赤外線データ 1 バイト受信/データ判定, LCD 表示を行う
リモコン送信初期化	sendir_init	リモコン送信処理の初期設定
1 バイト赤外線送信	sendir	1 ビットごとに赤外線データ変換
キースキャン	keyscan	選択キー判別, 1 バイト×10 回赤外線送信
ポート 5L 読み取り	keyread	P53~P50 の内容を戻す
タイマ C 割り込み	tcint	タイマ C による割り込み, フラグ設定
タイマ F 割り込み	tfint	タイマ FH による割り込み, P42 トグル出力による搬送波作成
タイマ G 割り込み	tgint	インプットキャプチャ開始, 測定, フラグ設定
LCD 初期化	lcd_init	LCDRAM の初期化

(2) 引数の説明

表 6 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
sdt	1 バイト送信データ	sendir	1 バイト	入力

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを表 7 に示します。

表 7 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMC	TMC7	タイマモードレジスタ C(オートリロード機能選択) : TMC7=0 のとき, タイマ C 機能をインターバル機能に設定 : TMC7=1 のとき, タイマ C 機能をオートリロード機能に設定	0xFFB4 ビット 7	1
	TMC6 TMC5	タイマモードレジスタ C(カウンタアップ/ダウン制御) : TMC6=0, TMC5=0 のとき, TCC はアップカウンタ : TMC6=0, TMC5=1 のとき, TCC はダウンカウンタ : TMC6=1, TMC5=x のとき, TCC は UD 端子入力によるハードウェア制御 (注 x:TMC5 は, 0, 1 どちらでも良い)	0xFFB4 ビット 6 ビット 5	TMC6=0 TMC5=0
	TMC2 TMC1 TMC0	タイマモードレジスタ C(クロックセレクト) : TMC2=0, TMC1=1, TMC0=1 のとき, TCC は, 内部クロック /64 でカウントする	0xFFB4 ビット 2 ビット 1 ビット 0	TMC2=0 TMC1=1 TMC0=1
TCC	タイマカウンタ C : システムクロックを 16 分周したクロックを入力とする 8 ビットのアップカウンタで, オーバフローすると TCC に TLC の設定値がロードされる	0xFFB5	0x00	

表7 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
TLC		タイマロードレジスタ C : TLC に値を設定すると, TCC は TLC 設定値からカウントアップを開始し, TCC がオーバーフローすると TCC に TLC の設定値がロードされる	0xFFB5	—
TCRF	CKSH2 CKSH1 CKSH0	タイマコントロールレジスタ F(クロックセレクト H) : CKSH2=1, CKSH1=1, CKSH0=0 のとき, TCFH は, 内部クロック /4 でカウントする	0xFFB6 ビット 6 ビット 5 ビット 4	CKSH2=1 CKSH1=1 CKSH0=0
TCSRFB	TFOVFH	タイマコントロールステータスレジスタ F (タイマオーバーフローフラグ H) : TFOVFH=0 のとき, TCF がオーバーフローしていない : TFOVFH=1 のとき, TCF がオーバーフローした	0xFFB7 ビット 7	0
	CMFH	タイマコントロールステータスレジスタ F (コンペアマッチフラグ H) : CMFH=0 のとき, コンペアマッチ F が発生していない : CMFH=1 のとき, コンペアマッチ F が発生した	0xFFB7 ビット 6	0
	OVIEH	タイマコントロールステータスレジスタ F (タイマオーバーフローインターラプトイネーブル H) : OVIEH=0 のとき, TCF のオーバーフローによる割り込み要求を禁止 : OVIEH=1 のとき, TCF のオーバーフローによる割り込み要求を許可	0xFFB7 ビット 5	0
	CCLRHB	タイマコントロールステータスレジスタ F (カウンタクリア H) : CCLRHB=0 のとき, コンペアマッチによる TCFH のクリアを禁止 : CCLRHB=1 のとき, コンペアマッチによる TCFH のクリアを許可	0xFFB7 ビット 4	1
TCFH		8 ビットタイマカウンタ FH : /4 を入力とする 8 ビットのアップカウンタ	0xFFB8	0x00
OCRFB		アウトプットコンペアレジスタ FH : TCFH と比較される。OCRFB と TCFH の値が一致すると, TCSRFB の CMFH が 1 にセットされる。	0xFFBA	0x80
PMR5		ポートモードレジスタ 5 (P5n/~WKPn/SEGn+1 端子機能切り換え) ~ SEGn+1 端子としない場合の端子機能 : PMR5=0x00 のとき, P5n/~WKPn/SEGn+1 端子は, P5n 入出力端子として機能	0xFFCC	0x00
PDR4	P42	ポートデータレジスタ 4(P42) : P42=0 のとき, P42 端子は Low レベル : P42=1 のとき, P42 端子は High レベル	0xFFD7 ビット 2	—

表 7 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
PDR5	PDR5H	ポートデータレジスタ 5(P57 ~ P54) ~ PDR5 の上位 4 ビットを PDR5H とし, 出力ポート P57 ~ P54 に格納するデータを設定する。P57 ~ P54 のデータのうち, Low レベルに設定したキー列の状態が, P53 ~ P50 に反映される。 : PDR5=0xE0 のとき, P54 のキー列(0, 4)を選択する。 : PDR5=0xD0 のとき, P55 のキー列(1, 5)を選択する。 : PDR5=0xB0 のとき, P56 のキー列(2, 6)を選択する。 : PDR5=0x70 のとき, P57 のキー列(3, 7)を選択する。	0xFFD8 ビット 7 ~ ビット 4	—
	PDR5L	ポート 7 レジスタ(P53 ~ P50) ~ PDR5 の下位 4 ビットを PDR5L とし, 入力ポート P53 ~ P50 の値が反映される。	0xFFD8 ビット 3 ~ ビット 0	—
PUCR5		ポートプルアップコントロールレジスタ 5 : PUCR5 = 0x00 のとき, プルアップ MOS を OFF にする。	0xFFE2	0x00
PCR4		ポートコントロールレジスタ 4 : PCR4=0xFC のとき, P42 を出力ポートに設定。	0xFFE8	0xFC
PCR5		ポートコントロールレジスタ 5 : PCR5=0xF0 のとき, P57 ~ P54 を出力ポート, P53 ~ P50 を入力ポートに設定。	0xFFE8	0xF0
TMG	TGOVFH	タイマモードレジスタ G(タイマオーバフローフラグ H) : TGOVFH=0 のとき, インพุットキャプチャ入力信号が High レベルで TCG がオーバフローしていない : TGOVFH=1 のとき, インพุットキャプチャ入力信号が High レベルで TCG がオーバフローした	0xFFBC ビット 7	0
	TGOVFL	タイマモードレジスタ G(タイマオーバフローフラグ L) : TGOVFL=0 のとき, インพุットキャプチャ入力信号が Low レベルで TCG がオーバフローしていない : TGOVFL=1 のとき, インพุットキャプチャ入力信号が Low レベルで TCG がオーバフローした	0xFFBC ビット 6	0
	OVIE	タイマモードレジスタ G (タイマオーバフローインタラプトイネーブル) : OVIE=0 のとき, TCG のオーバフローによる割り込み要求を禁止 : OVIE=1 のとき, TCG のオーバフローによる割り込み要求を許可	0xFFBC ビット 5	0
	IIEGS	タイマモードレジスタ G (インพุットキャプチャインタラプトエッジセレクト) : IIEGS=0 のとき, インพุットキャプチャ入力信号の立ち上がりエッジで割り込みを発生 : IIEGS=1 のとき, インพุットキャプチャ入力信号の立ち下がりエッジで割り込みを発生	0xFFBC ビット 4	0

表 7 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
TMG	CCLR1 CCLR0	タイマモードレジスタ G(カウンタクリア 1, 0) : CCLR1=1, CCLR0=0 のとき, インพุットキャプチャ入力信号の立ち上がりエッジにより TCG をクリア	0xFFBC ビット 3 ビット 2	CCLR1=1 CCLR0=0
	CKS1 CKS0	タイマモードレジスタ G(クロックセレクト 1, 0) : CKS1=0, CKS0=0 のとき TCG の入力カウントを /64 に設定	0xFFBC ビット 1 ビット 0	CKS1=0 CKS0=0
TCG		タイマカウンタ G ~8 ビット, リード/ライト不可のレジスタで, 入力したクロックによりカウントアップする。TCG は, インพุットキャプチャ入力信号の立ち上がりエッジが検出されると, そのときの TCG の値がインพุットキャプチャレジスタ GR(ICRGR)に転送される。	—	—
ICRGR		インพุットキャプチャレジスタ GR ~8 ビットのリード専用レジスタで, インพุットキャプチャ入力信号の立ち上がりエッジが検出されると, そのときの TCG の値が転送される。	0xFFBE	—
LPCR	DTS1 DTS0	LCD ポートコントロールレジスタ(デューティ比選択 1, 0) ~ DTS1, DTS0 の組み合わせで, スタティック, 1/2 ~ 1/4 デューティのいずれかを選択します。 : DTS1=1, DTS0=1 のとき, 1/4 デューティを選択します	0xFFC0 ビット 7 ビット 6	DTS1=1 DTS0=1
	CMX	LCD ポートコントロールレジスタ(コモン機能選択) ~ デューティによって使用しないコモン端子をコモンドライバ能力を大きくするために複数の端子から同じ波形を出力するか否かを選択します。 : CMX=0 のとき, デューティによって使用しない複数のコモン端子から同じ波形を出力しない : CMX=1 のとき, デューティによって使用しない複数のコモン端子から同じ波形を出力する	0xFFC0 ビット 5	0
	SGS3 SGS2 SGS1 SGS0	LCD ポートコントロールレジスタ (セグメントドライバ選択 3 ~ 0) ~ 使用するセグメントドライバを選択します。 : SGS3=1, SGS2=0, SGS1=1, SGS0=1 のとき, SEG32 ~ SEG13 端子はセグメントドライバとして, SEG12 ~ SEG1 端子はポートとして機能	0xFFC0 ビット 3 ビット 2 ビット 1 ビット 0	SGS3=1 SGS2=0 SGS1=1 SGS0=1
LCR	PSW	LCD コントロールレジスタ(LCD 電源分割抵抗接続制御) ~ 低消費電力モードで LCD 表示をしない場合, また外部電源を使用する場合に LCD 電源分割抵抗を Vcc から切断することができます。ACT=0 とした場合, またスタンバイモード時には本ビットとは無関係に LCD 電源分割抵抗が Vcc から切断されます。 : PSW=0 のとき, LCD 電源分割抵抗を Vcc から切断 : PSW=1 のとき, LCD 電源分割抵抗を Vcc に接続	0xFFC1 ビット 6	0

表 7 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
LCR	ACT	LCD コントロールレジスタ(表示機能開始) ~ LCD コントローラ/ドライバを使用するかしないかを選択 します。本ビットを"0"にクリアすることにより、LCD コ ントローラ/ドライバは動作を停止します。また、PSW の値 と無関係に LCD 駆動電源が OFF 状態となります。ただし、 レジスタの内容は保持されます。 : ACT=0 のとき、LCD コントローラ/ドライバは動作停止 : ACT=1 のとき、LCD コントローラ/ドライバは動作	0xFFC1 ビット 5	1
	DISP	LCD コントロールレジスタ(表示データ制御) ~ DISP は LCDRAM の内容を表示するか、LCDRAM の内容 に関係なくブランクデータを表示するかを選択します。 : DISP=0 のとき、ブランクデータを表示 : DISP=1 のとき、LCDRAM データを表示	0xFFC1 ビット 4	1
	CKS3 CKS2 CKS1 CKS0	LCD コントロールレジスタ(フレーム周波数選択 3~0) ~ 使用クロックの選択とフレーム周波数の選択を行います。 : CKS3=1, CKS2=1, CKS1=1, CKS0=0 のとき、使用クロッ クに /128 を選択	0xFFC1 ビット 3 ビット 2 ビット 1 ビット 0	CKS3=1 CKS2=1 CKS1=1 CKS0=0
LCR2	LCDAB	LCD コントロールレジスタ 2(A 波形/B 波形切り換えの制御) ~ LCD の駆動波形を A 波形にするか B 波形にするかを選択 します。 : LCDAB=0 のとき、LCD は A 波形で駆動 : LCDAB=1 のとき、LCD は B 波形で駆動	0xFFC2 ビット 7	0
LCDRAM		LCDRAM ~ LCD の表示データを設定します。	0xF740 ~ 0xF74F	—
PMR1	TMIG	ポートモードレジスタ 1(P13/TMIG 端子機能切り替え) : TMIG=0 のとき、P13/TMIG 端子は P13 入出力端子として 機能 : TMIG=1 のとき、P13/TMIG 端子は TMIG 入力端子として 機能	0xFFC8 ビット 3	1
PMR2	NCS	ポートモードレジスタ 2(TMIG ノイズキャンセラセレクト) : NCS=0 のとき、インプットキャプチャ入力信号のノイズ 除去機能なし : NCS=1 のとき、インプットキャプチャ入力信号のノイズ 除去機能あり	0xFFC9 ビット 1	0
IENR2	IENTG	割り込み許可レジスタ 2(タイマ G 割り込みイネーブル) ~ タイマ G 割り込み要求の許可/禁止を制御します。 : IENTG=0 のとき、タイマ G 割り込み要求を禁止 : IENTG=1 のとき、タイマ G 割り込み要求を許可	0xFFFF4 ビット 4	1
	IENFHH	割り込み許可レジスタ 2(タイマ FH 割り込みイネーブル) : IENFHH=0 のとき、タイマ FH 割り込み要求を禁止 : IENFHH=1 のとき、タイマ FH 割り込み要求を許可	0xFFFF4 ビット 3	1
	IENTC	割り込み許可レジスタ 2(タイマ C 割り込みイネーブル) : IENTC=0 のとき、タイマ C 割り込み要求を禁止 : IENTC=1 のとき、タイマ C 割り込み要求を許可	0xFFFF4 ビット 1	1

表 7 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
IRR2	IRRTG	割り込み要求レジスタ 2(タイマ G 割り込み要求フラグ) ~ タイマ G 割り込み要求の有無を反映します。 : IRRTG=0 のとき, タイマ G 割り込みが要求されていない : IRRTG=1 のとき, タイマ G 割り込みが要求されている	0xFFFF7 ビット 4	0
	IRRTFH	割り込み要求レジスタ 2(タイマ FH 割り込み要求フラグ) : IRRTFH=0 のとき, タイマ FH 割り込みが要求されていない : IRRTFH=1 のとき, タイマ FH 割り込みが要求されている	0xFFFF7 ビット 3	0
	IRRTC	割り込み要求レジスタ 2(タイマ C 割り込み要求フラグ) : IRRTC=0 のとき, タイマ C 割り込みが要求されていない : IRRTC=1 のとき, タイマ C 割り込みが要求されている	0xFFFF7 ビット 1	0

(4) 使用 RAM 説明

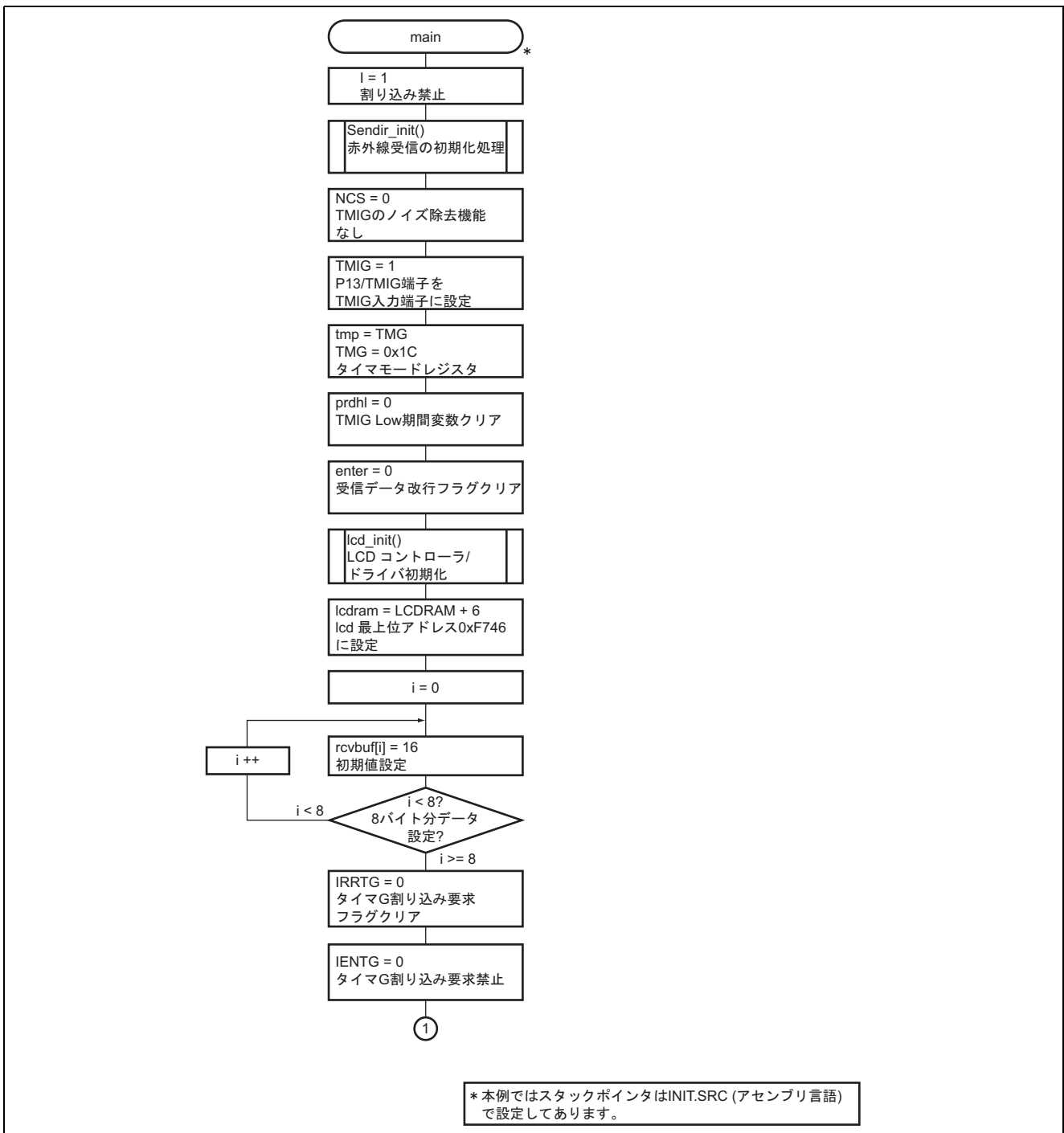
本タスク例の使用 RAM を表 8 に示します。

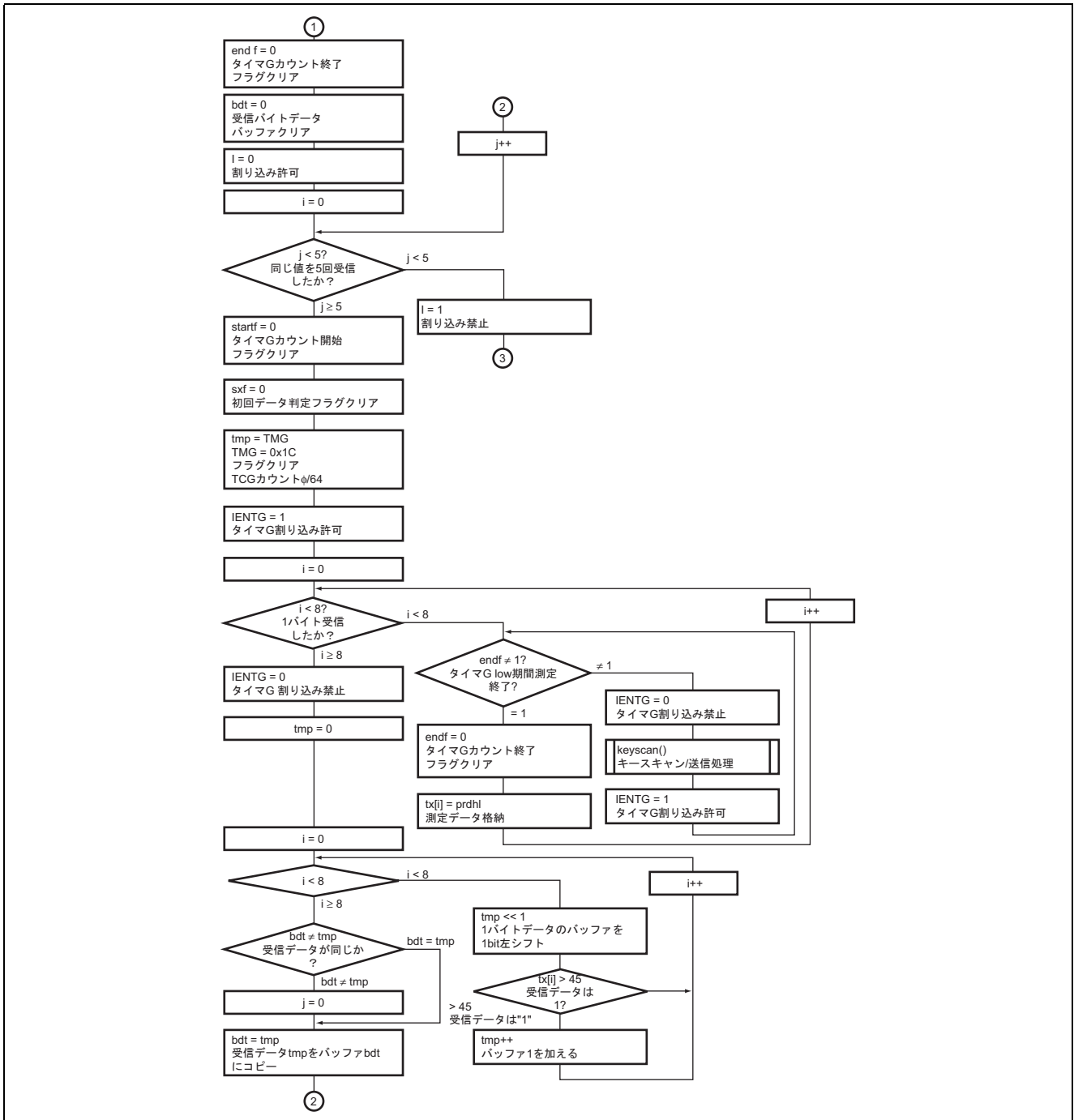
表 8 使用 RAM 説明

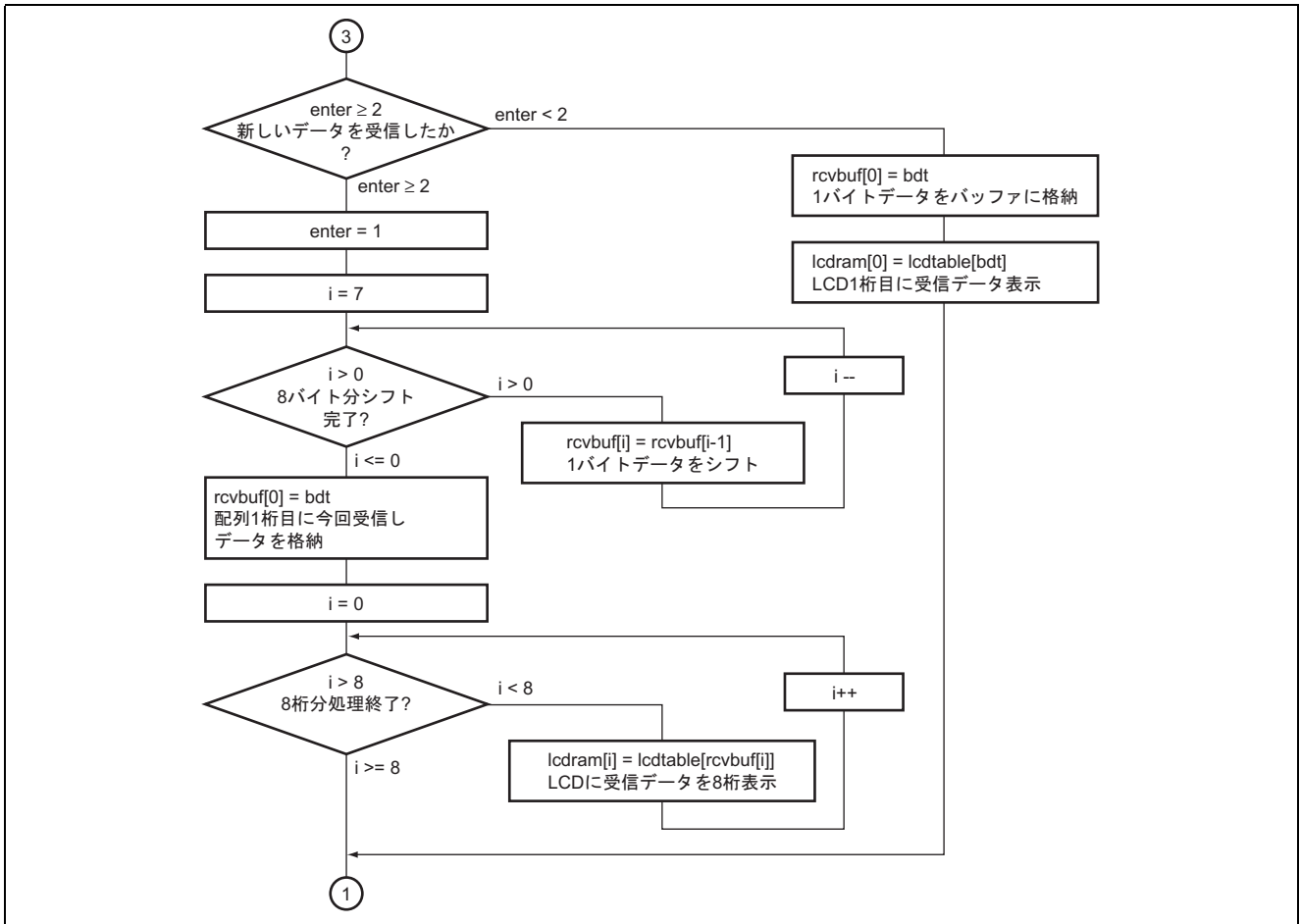
ラベル名	機能	メモリ消費量	使用モジュール名
tcflg	タイマ C 割り込み判定	1 バイト	main,tcint,sendir
prdhl	タイマ G 測定結果を格納する	1 バイト	main,tgint
sxf	新しいデータか同じデータかを判定する	1 バイト	main,tgint
enter	LCD を更新するかどうかを判定する	1 バイト	main,tgint
startf	2 回目のタイマ G 割り込みか否かを判定するフラグ	1 バイト	main,tgint
endf	周期測定が終了したか否かを判定するフラグ	1 バイト	main,tgint

6. フローチャート

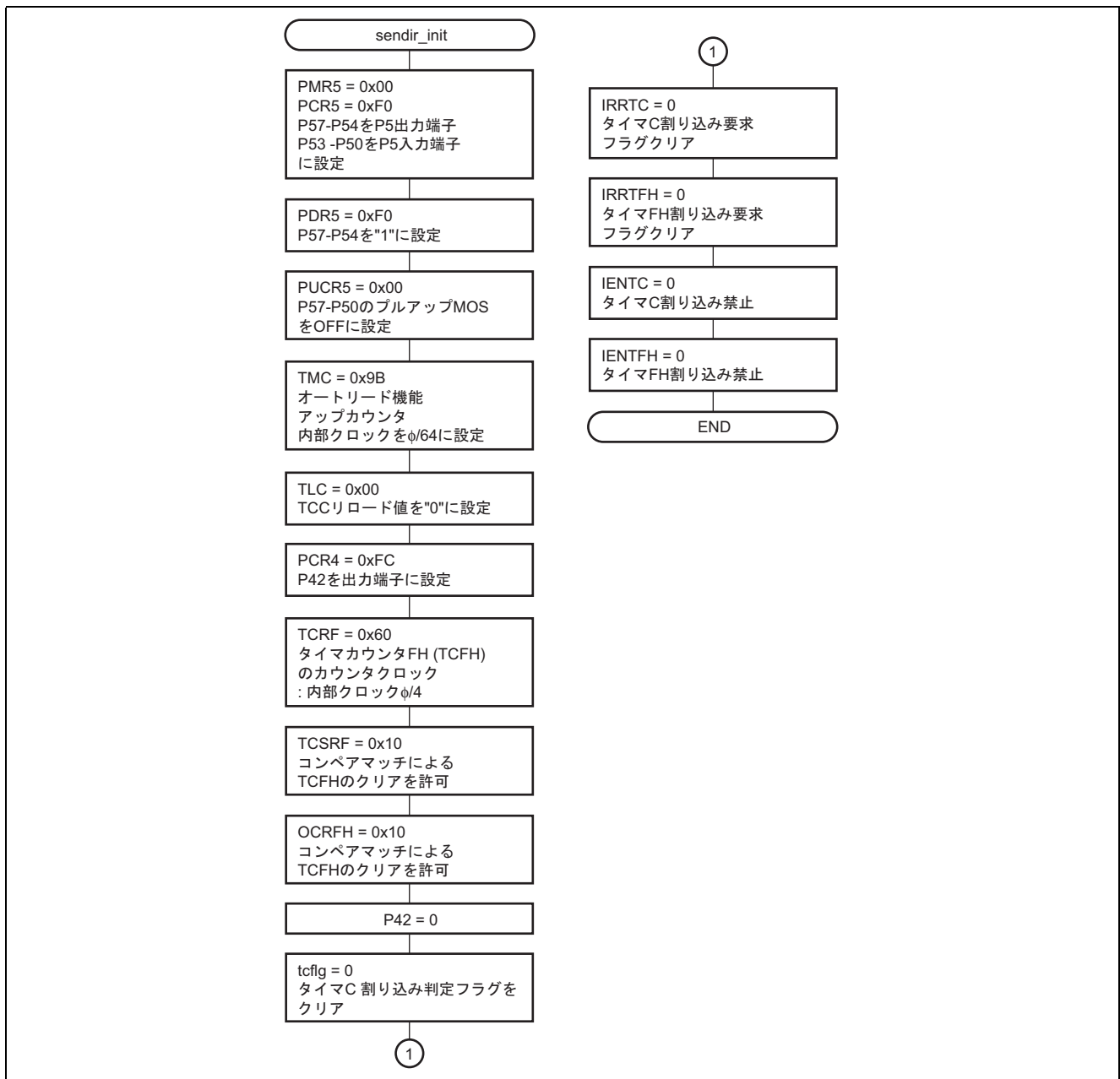
(1) メインルーチン



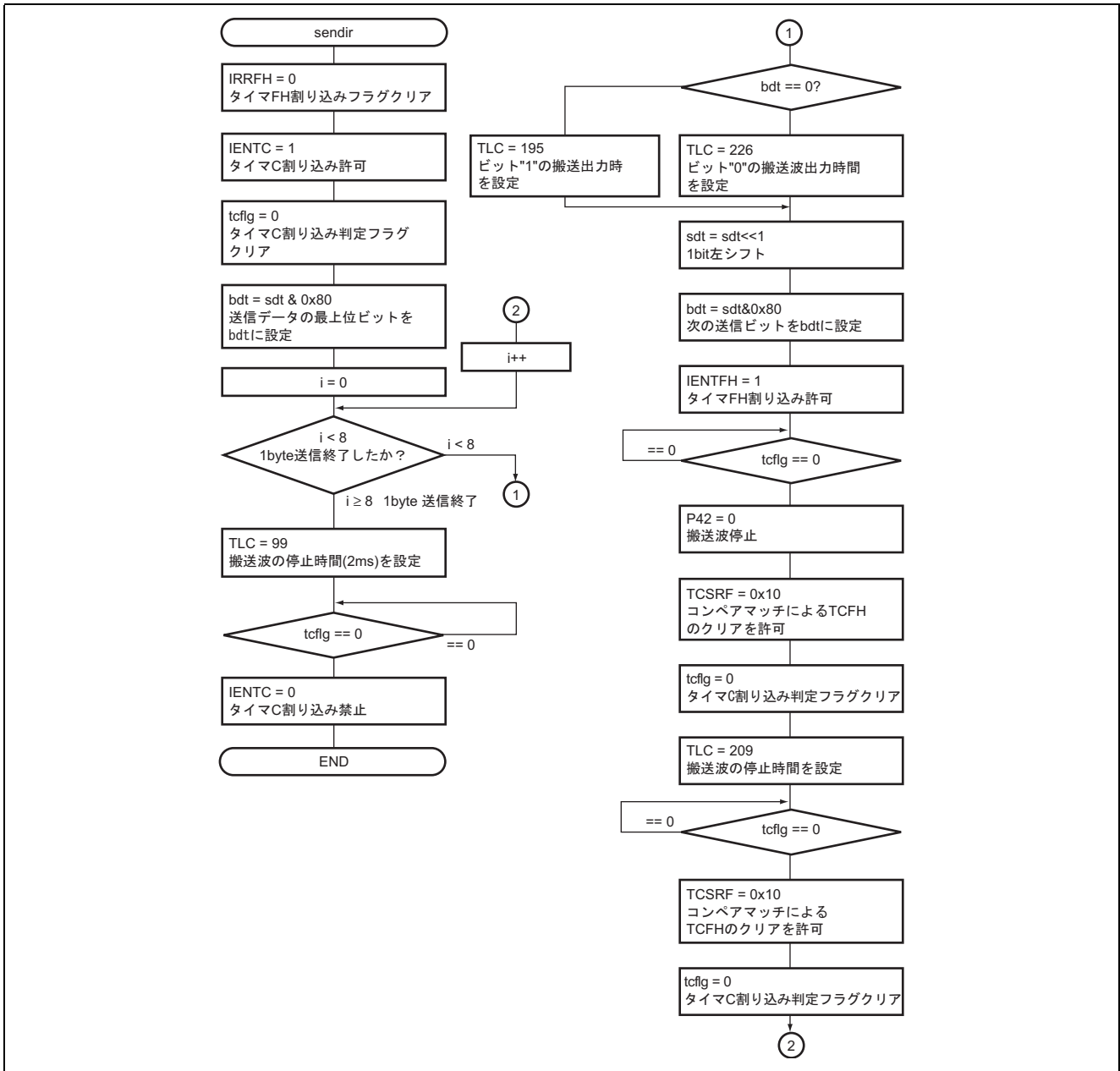




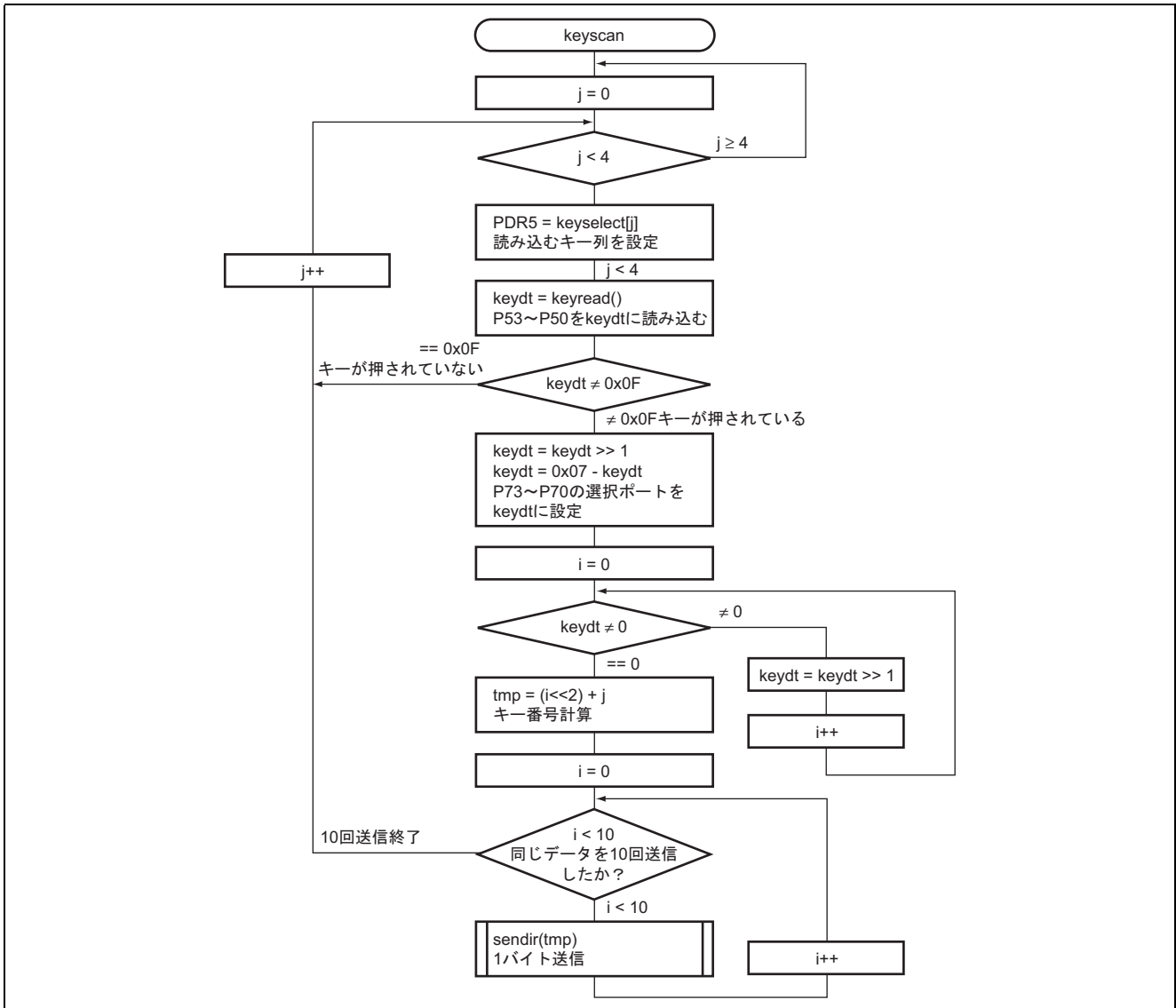
(2) リモコン送信初期化



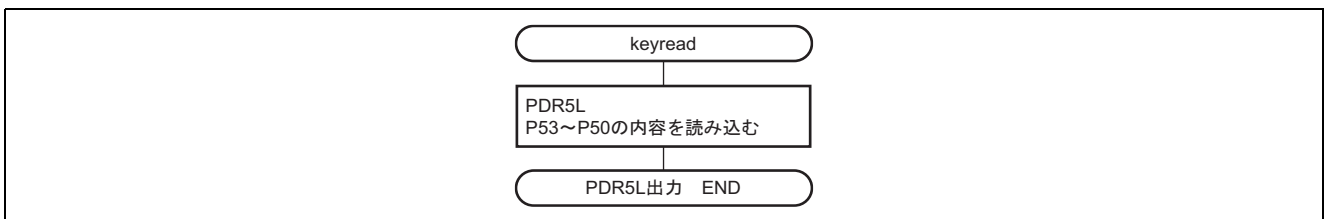
(3) リモコン送信処理



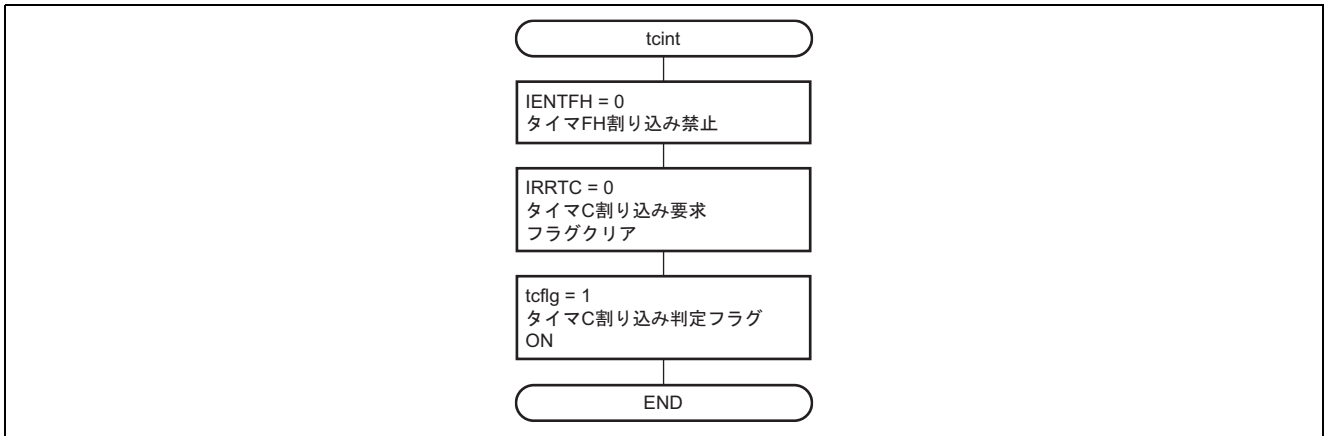
(4) キースキャン



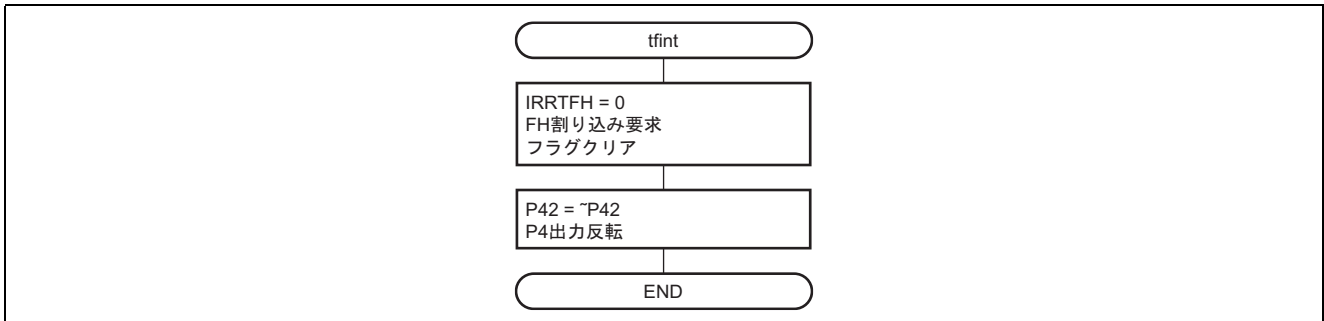
(5) ポート 5L 読み取り



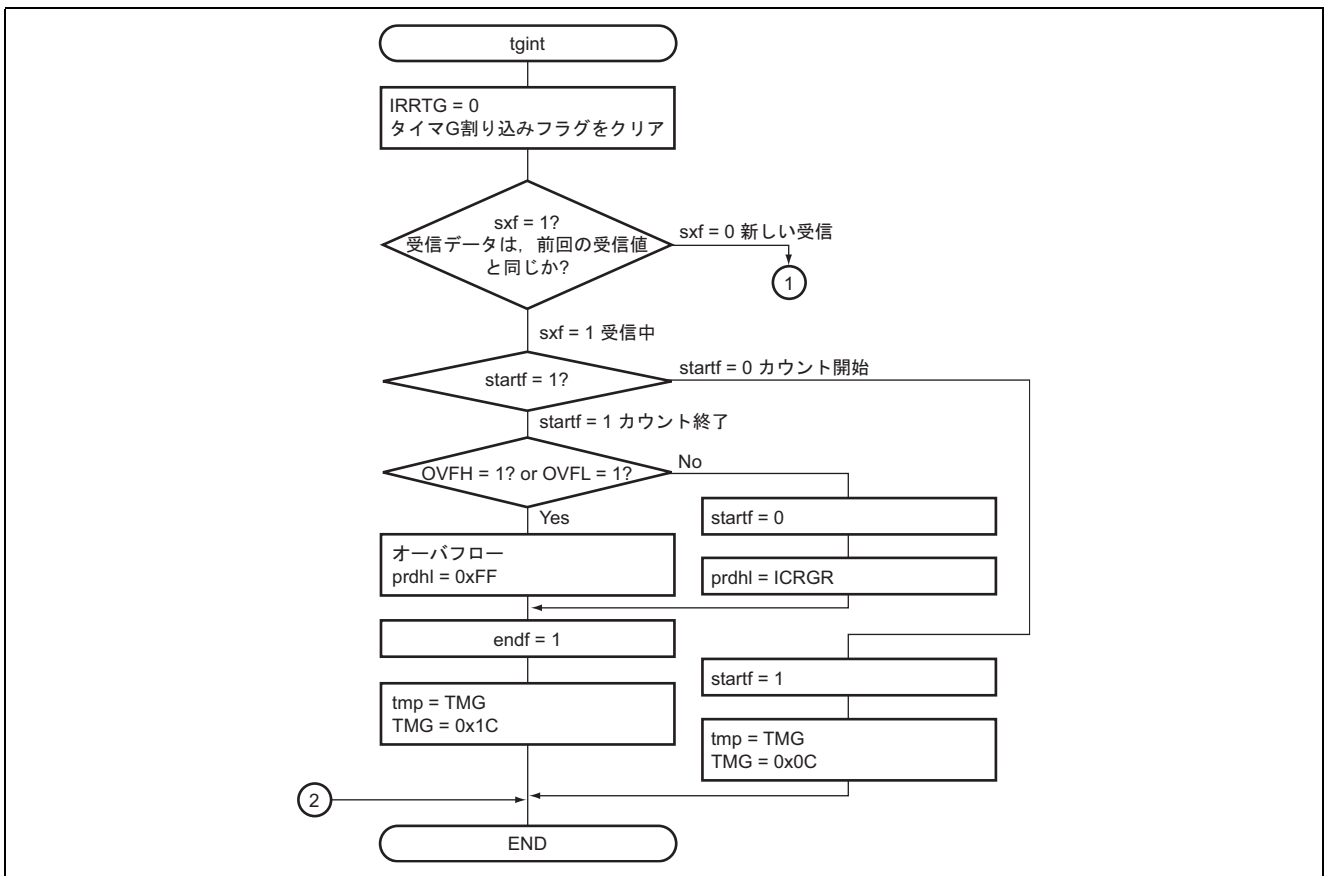
(6) タイマ C 割り込み

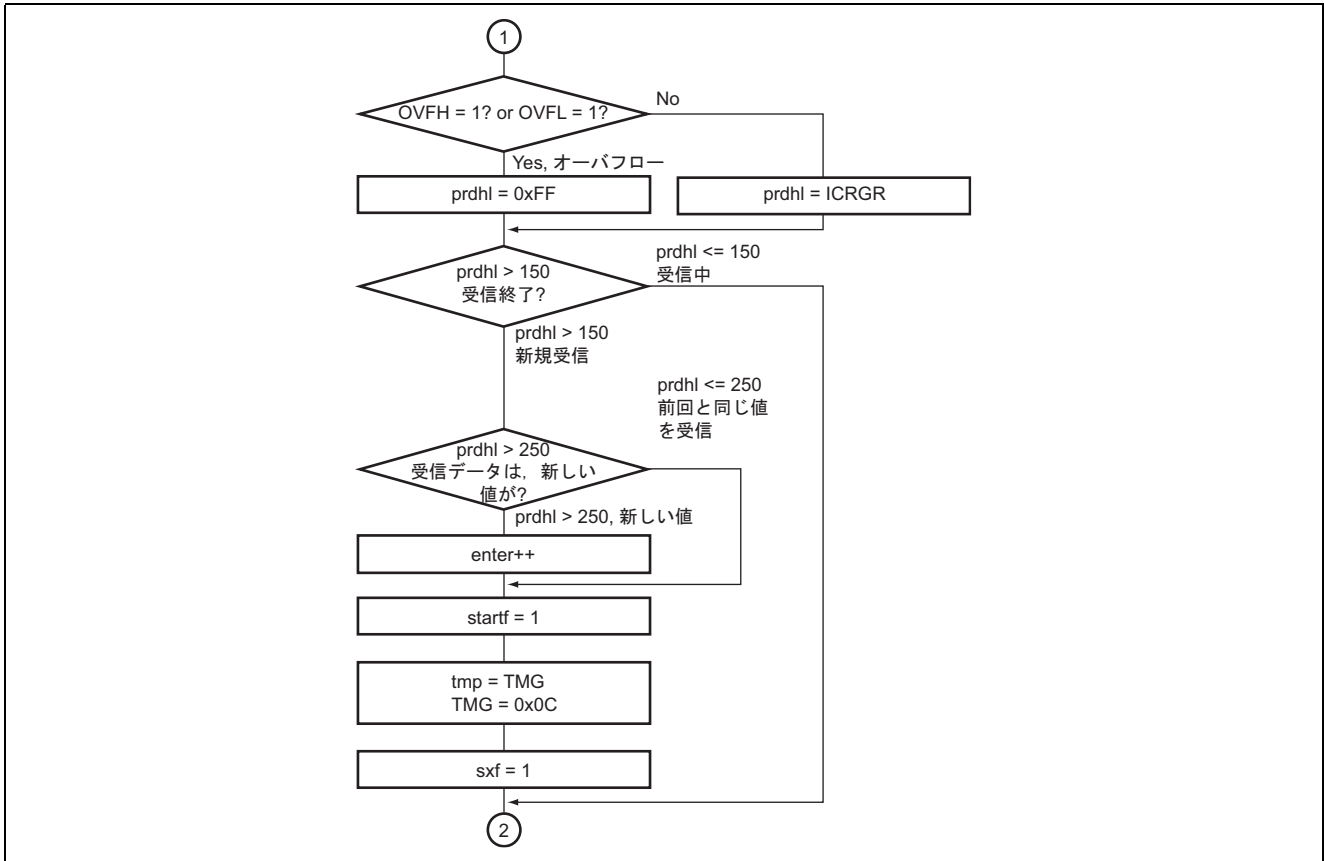


(7) タイマ F 割り込み

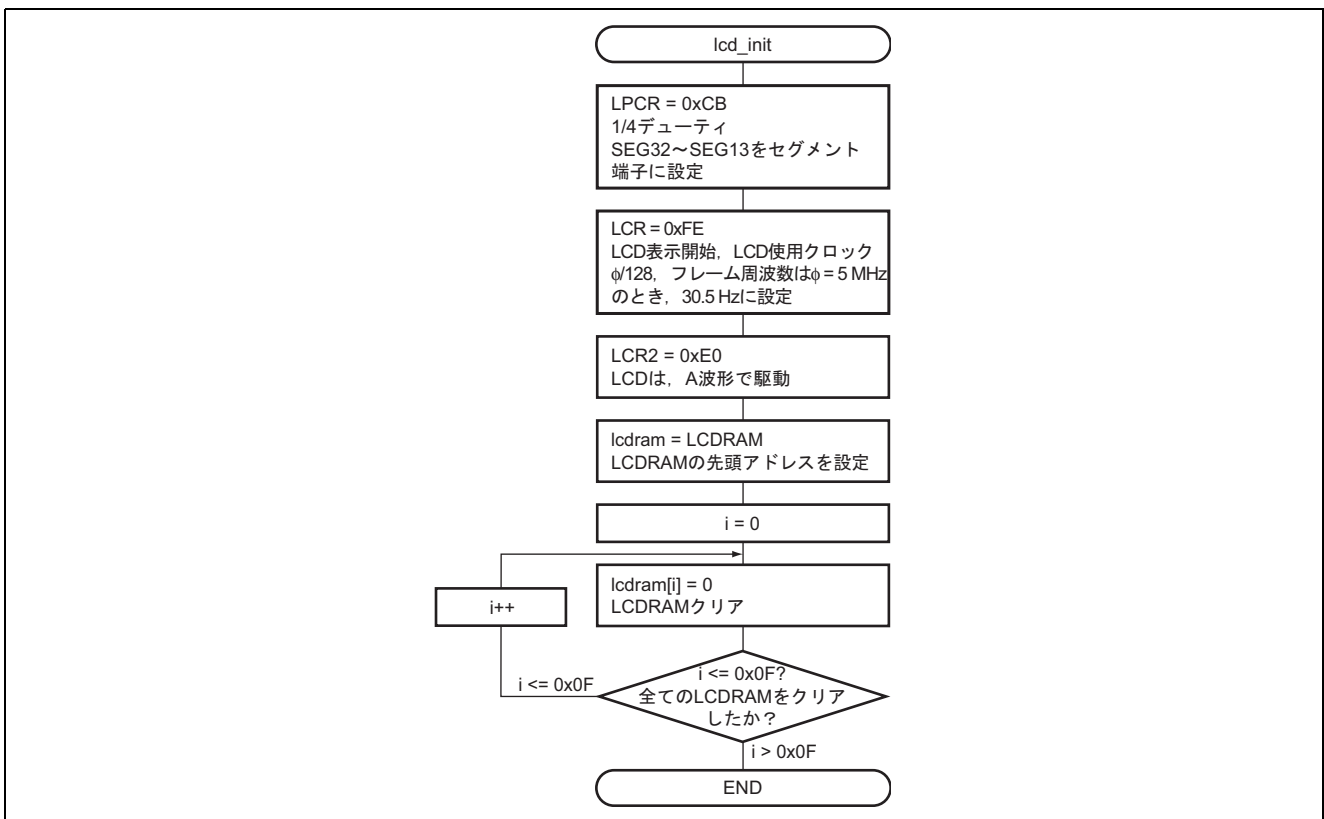


(8) タイマ G 割り込み





(9) LCD 初期化



7. プログラムリスト

7.1 INIT.SRC(プログラムリスト)

```

        .export _INIT
        .import _main
;
        .section    P, CODE
        _INIT:
        mov.w    #h'ff80, r7
        ldc.b    #b'10000000, ccr
        jmp @_main
;
        .end

/*****/
/*                                     */
/* H8/300L Super Low Power Series      */
/*   -H8/38024 Series-                 */
/* Application Note                     */
/*                                     */
/* 'Infrared radiation Send/Receive Function' */
/*                                     */
/* Function                             */
/* :Timer C Auto-reload Timer          */
/* :Timer FH 8-bit Timer                */
/* :Timer G Input capture Timer        */
/*                                     */
/* External Clock : 10MHz               */
/* Internal Clock : 5MHz                */
/* Sub Clock      : 32.768kHz           */
/*                                     */
/*****/

#include    <machine.h>

/*****/
/* Symbol Definition                    */
/*****/
struct BIT {
    unsigned char    b7:1;    /* bit7 */
    unsigned char    b6:1;    /* bit6 */
    unsigned char    b5:1;    /* bit5 */
    unsigned char    b4:1;    /* bit4 */
    unsigned char    b3:1;    /* bit3 */
    unsigned char    b2:1;    /* bit2 */
    unsigned char    b1:1;    /* bit1 */
    unsigned char    b0:1;    /* bit0 */
};

struct P4BIT {
    unsigned char    H:4;    /* bit7-bit4 */
    unsigned char    L:4;    /* bit3-bit0 */
};

#define    TMC        *(volatile unsigned char *)0xFFB4 /* Timer Mode Register C */
#define    TCC        *(volatile unsigned char *)0xFFB5 /* Timer Counter C */

```



```

#define TLC      *(volatile unsigned char *)0xFFB5 /* Timer Load Register C */
#define TCRF     *(volatile unsigned char *)0xFFB6 /* Timer Control Register F */
#define TCRF_BIT (*(struct BIT *)0xFFB6) /* Timer Control Register F */
#define TOLH     TCRF_BIT.b7 /* Toggle Output Level F */
#define CKSH2    TCRF_BIT.b6 /* Clock Select H2 */
#define CKSH1    TCRF_BIT.b5 /* Clock Select H1 */
#define CKSH0    TCRF_BIT.b4 /* Clock Select H0 */
#define TCSRFB   *(volatile unsigned char *)0xFFB7 /* Timer Control Status Register F */
#define TCSRFB_BIT (*(struct BIT *)0xFFB7) /* Timer Control Status Register F */
#define TFOVFH   TCSRFB_BIT.b7 /* Timer Overflow Flag H */
#define CMFH     TCSRFB_BIT.b6 /* Compare Match Flag H */
#define OVIEH    TCSRFB_BIT.b5 /* Timer Overflow Interrupt Enable H*/
#define CCLRHR   TCSRFB_BIT.b4 /* Counter clear H */
#define TCFH     *(volatile unsigned char *)0xFFB8 /* Timer Counter FH */
#define OCRFH    *(volatile unsigned char *)0xFFBA /* Output Compare Register FH */
#define PMR5     *(volatile unsigned char *)0xFFCC /* Port mode register 5 */
#define PDR4_BIT (*(struct BIT *)0xFFD7) /* Port data register 4 */
#define P42      PDR4_BIT.b2 /* P42 */
#define PDR5     *(volatile unsigned char *)0xFFD8 /* Port data register 5 */
#define PDR5_BIT (*(struct P4BIT *)0xFFD8) /* Port mode register 5 */
#define PDR5H    PDR5_BIT.H /* P57-P54 */
#define PDR5L    PDR5_BIT.L /* P53-P50 */
#define PUCR5    *(volatile unsigned char *)0xFFE2 /* Port pull-up control register 5 */
#define PCR4     *(volatile unsigned char *)0xFFE7 /* Port control register 4 */
#define PCR5     *(volatile unsigned char *)0xFFE8 /* Port control register 5 */
#define LCDRAM   (volatile unsigned char *)0xF740 /* LCD RAM */
#define TMG      *(volatile unsigned char *)0xFFBC /* Timer Mode Register G */
#define TMG_BIT  (*(struct BIT *)0xFFBC) /* Timer Mode Register G */
#define TGOVFH   TMG_BIT.b7 /* Timer Overflow Flag H */
#define TGOVFL   TMG_BIT.b6 /* Timer Overflow Flag L */
#define OVIE     TMG_BIT.b5 /* Timer Overflow Interrupt Enable */
#define IIEGS    TMG_BIT.b4 /* Input Capture Interrupt Edge Select */
#define CCLR1    TMG_BIT.b3 /* Counter Clear 1 */
#define CCLR0    TMG_BIT.b2 /* Counter Clear 0 */
#define CKS1     TMG_BIT.b1 /* Clock Select 1 */
#define CKS0     TMG_BIT.b0 /* Clock Select 0 */
#define ICRGF    *(volatile unsigned char *)0xFFBD /* Input Capture Register GF */
#define ICRGR    *(volatile unsigned char *)0xFFBE /* Input Capture Register GR */
#define LPCR     *(volatile unsigned char *)0xFFC0 /* LCD Port Control Register */
#define LCR      *(volatile unsigned char *)0xFFC1 /* LCD Control Register */
#define LCR2     *(volatile unsigned char *)0xFFC2 /* LCD Control Register 2 */
#define PMR1     *(volatile unsigned char *)0xFFC8 /* Port Mode Register 1 */
#define PMR1_BIT (*(struct BIT *)0xFFC8) /* Port Mode Register 1 */
#define TMIG     PMR1_BIT.b3 /* P13/TMIG Input Select */
#define PMR2     *(volatile unsigned char *)0xFFC9 /* Port Mode Register 2 */
#define PMR2_BIT (*(struct BIT *)0xFFC9) /* Port Mode Register 2 */
#define NCS      PMR2_BIT.b1 /* TMIG noise canceller select */
#define IENR2    *(volatile unsigned char *)0xFFF4 /* Interrupt Enable Register 2 */
#define IENR2_BIT (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 2 */
#define IENTG    IENR2_BIT.b4 /* Timer G Interrupt Enable */
#define IENTFH   IENR2_BIT.b3 /* Timer FH Interrupt Enable */
#define IENTC    IENR2_BIT.b1 /* Timer C Interrupt Enable */
#define IRR2_BIT (*(struct BIT *)0xFFF7) /* Interrupt Request Register 2 */
#define IRRTG    IRR2_BIT.b4 /* Timer G Interrupt Request Flag */
#define IRRTFH   IRR2_BIT.b3 /* Timer FH Interrupt Request Flag */
#define IRRTC    IRR2_BIT.b1 /* Timer C Interrupt Request Flag */
    
```

```

#pragma interrupt (tcint)
#pragma interrupt (tfint)
#pragma interrupt (tgint)
/*****
/* Function define */
*****/
extern void INIT( void ); /* SP Set */
void main( void );
void sendir_init( void );
void sendir( unsigned char sdt );
void keyscan( void );
unsigned char keyread( void );
void tcint( void );
void tfint( void );
void tgint( void );
void lcd_init( void );

/*****
/* RAM define */
*****/
volatile unsigned char tcflg; /* Timer C Interrupt Flag */
unsigned char prdhl,sxf,enter; /* Timer G Interrupt Flag */
unsigned char startf; /* Timer G Count Start Flag */
unsigned char endf; /* Timer G Count End Flag */

unsigned char keyselect[4] = {
    0xE0,
    0xD0,
    0xB0,
    0x70,
};

unsigned char lcdtable[17] = { /* LCD Key Select Table */
    0xD7, /* 0 */
    0x06, /* 1 */
    0xE3, /* 2 */
    0xA7, /* 3 */
    0x36, /* 4 */
    0xB5, /* 5 */
    0xF5, /* 6 */
    0x07, /* 7 */
    0xF7, /* 8 */
    0xB7, /* 9 */
    0x77, /* A */
    0xF4, /* B */
    0xD1, /* C */
    0xE6, /* D */
    0xF1, /* E */
    0x71, /* F */
    0x00, /* */
};

/*****
/* Vector Address */
*****/
#pragma section V1 /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = { /* 0x00 - 0x0f */

```

```

INIT                                     /* 00 Reset                               */
};
#pragma section      V2                  /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL2[])(void) = {
    tcint                                /* 1A Timer C Interrupt                    */
};
#pragma section      V3                  /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL3[])(void) = {
    tfint                                /* 1E Timer F Interrupt                    */
};
#pragma section      V4                  /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL4[])(void) = {
    tgint                                /* 20 Timer G Interrupt                    */
};

#pragma section                          /* P                                         */
/*****/
/* Main Program                          */
/*****/
void main( void )
{
    unsigned char i,j,tx[8],rcvbuf[8],tmp,bdt;
    unsigned char *lcdram;

    set_imask_ccr(1);                    /* Interrupt Disable                        */

    sendir_init();

    NCS = 0;                             /* No noise cancellation circuit          */
    TMIG = 1;                             /* P13/TMIG input select                  */
    tmp = TMG;                             /* Dummy Read for Flag Clear              */
    TMG = 0x1C;                             /* TMG Set                                 */

    prdhl = 0;                             /* Caputure Data Ram Clear                */
    enter = 0;                             /* Enter flag Clear                       */

    lcd_init();                            /* Initialize LCD                          */
    lcdram = LCDRAM + 0x0006;              /* Set LCDRAM Address                     */
    for(i = 0; i < 8; i++){
        rcvbuf[i] = 16;
    }

    IRRTG = 0;                             /* Clear IRRTG                            */
    IENTG = 0;                             /* Timer G Interrupt Disable              */

    while(1){
        endf = 0;                          /* Timer G Interrupt EndFlag Clear        */
        bdt = 0;                            /* Data Buffer Clear                       */

        set_imask_ccr(0);                  /* Interrupt Enable                        */
        for(j = 0; j < 5; j++){
            startf = 0;                    /* Timer G Interrupt StartFlag Clear*/
            sxf = 0;                       /* Flag Clear                             */

            tmp = TMG;                     /* Dummy Read for Flag Clear              */
            TMG = 0x1C;                    /* Timer Mode Register Set                */
            IENTG = 1;                     /* Timer G Interrupt Enable               */
        }
    }
}

```

```

        for(i = 0; i < 8; i++){
            while(endf != 1){
                IENTG = 0;
                keyscan();
                IENTG = 1;
            }
            endf = 0;
            tx[i] = prdhl;
        }
        IENTG = 0;

        tmp = 0;
        for(i = 0; i < 8; i++){
            tmp = tmp<<1;
            if(tx[i] > 45){
                tmp++;
            }
        }

        if(bdt != tmp){
            j = 0;
        }

        bdt = tmp;
    }
    set_imask_ccr(1);

    if(enter >= 2){
        enter = 1;
        for(i = 7; i > 0; i--){
            rcvbuf[i] = rcvbuf[i-1];
        }
        rcvbuf[0] = bdt;

        for(i = 0; i < 8; i++){
            lcdram[i] = lcdtable[rcvbuf[i]];
        }
    }
    else{
        rcvbuf[0] = bdt;
        lcdram[0] = lcdtable[bdt];
    }
}

/*****
/* Infrared radiation Send Initialize
*****/
void sendir_init( void )
{
    PMR5 = 0x00;
    PCR5 = 0xF0;
    PDR5 = 0xF0;
    PUCR5 = 0x00;

    TMC = 0x9B;
    TLC = 0x00;

```

```

PCR4 = 0xFC; /* Set P42 Output Pin */

TCRF = 0x60; /* Select TimerFH , phi/4 */
TCSRFB = 0x10; /* TCFH clearing by compare match */
OCRFB = 0x10; /* Set Interrupt time is 26us */

P42 = 0; /* P42 Output Low level */
tcflg = 0;

IRRTC = 0; /* Clear IRRTC */
IRRTFB = 0; /* Clear IRRTFB */
IENTC = 0; /* Timer C Interrupt Disable */
IENTFB = 0; /* Timer FB Interrupt Disable */
}

/*****/
/* Infrared radiation Send */
/*****/
void sendir( unsigned char sdt )
{
    unsigned char bdt,i;

    IRRTFB = 0; /* Clear IRRTFB */
    IENTC = 1; /* Timer C Interrupt Enable */

    tcflg = 0;
    bdt = sdt&0x80; /* Set Send top bit to bdt */
    for(i = 0; i < 8; i++){ /* Send 1 byte */
        if(bdt == 0)
            TLC = 226; /* Set bit-0 output time */
        else
            TLC = 195; /* Set bit-1 output time */

        sdt = sdt<<1; /* Set Next bit to bdt */
        bdt = sdt&0x80;
        IENTFB = 1; /* Timer FB Interrupt Enable */
        while(tcflg == 0); /* 1-bit is Sending */
        P42 = 0; /* P42 is Low level */
        TCSRFB = 0x10; /* Initialize Overflow Interrupt */

        tcflg = 0;
        TLC = 209; /* Set P42 Low level time */
        while(tcflg == 0);
        TCSRFB = 0x10; /* Initialize Overflow Interrupt */
        tcflg = 0;
    }

    TLC = 99; /* Wait Over 2ms */
    while(tcflg == 0);

    IENTC = 0; /* Timer C Interrupt Disable */
}

/*****/
/* KeyScan */
/*****/
void keyscan( void )

```

```

{
    unsigned char tmp,i,j,keydt;

    for(j = 0; j < 4; j++){
        PDR5 = keyselect[j];                /* Set Key Select          */

        keydt = keyread();
        if(keydt != 0x0F){                  /* Touch Key?             */
            keydt = keydt>>1;              /* What Key?              */
            keydt = 0x07 - keydt;
            for(i = 0; keydt != 0; i++){
                keydt = keydt>>1;
            }

            i = i<<2;
            tmp = i+j;                      /* Set KeyNo -> tmp      */

            for(i = 0; i < 10; i++){        /* Same Data Output 10 time */
                sendir(tmp);               /* Send lbyte data to IR  */
            }
        }
    }
}

/*****
/* KeyRead                               */
*****/
unsigned char keyread( void )
{
    return(PDR5L);
}

/*****
/* Timer C Interrupt                       */
*****/
void tcint( void )
{
    IENTFH = 0;                            /* Timer FH Interrupt Disable */
    IRRTC = 0;                             /* Clear IRRTC                */
    tcflg = 1;                             /* Timer C Interrupt flag Set */
}

/*****
/* Timer F Interrupt                       */
*****/
void tfint( void )
{
    IRRTFH = 0;                            /* Clear IRRFH                */
    P42 = ~P42;                            /* Toggle Output P42          */
}

/*****
/* Timer G Interrupt                       */
*****/
void tgint( void )
{
    unsigned char tmp;

```

```

IRRTG = 0; /* Clear IRRTG */

if(sxf == 1){
    if(startf == 1){
        if((TGOVFH == 1)|(TGOVFL == 1)){
            prdhl = 0xFF;
        }
        else{
            startf = 0; /* Clear startf */
            prdhl =ICRGR; /* Caputure Data Ramcopy */
        }
        endf = 1; /* Set endf */
        tmp = TMG; /* Dummy Read for Flag Clear */
        TMG = 0x1C; /* Overflow Interrupt Disable */
    }
    else{
        startf = 1; /* Set startf */
        tmp = TMG; /* Dummy Read for Flag Clear */
        TMG = 0x0C; /* Timer Mode Register Set */
    }
}
else{
    if((TGOVFH == 1)|(TGOVFL == 1)){
        prdhl = 0xFF; /* Not Recive Data */
    }
    else{
        prdhl =ICRGF; /* Caputure Data Ramcopy */
    }

    if(prdhl > 150){
        if(prdhl > 250){
            enter++; /* First Data or Renew Data */
        }
        startf = 1; /* Set startf */
        tmp = TMG; /* Dummy Read for Flag Clear */
        TMG = 0x0C; /* Timer Mode Register Set */
        sxf = 1;
    }
}

/*****/
/* LCD Initialize */
/*****/
void lcd_init( void )
{
    unsigned char i;
    unsigned char *lcdram;

    LPCR = 0xCB; /* 1/4 Duty ,Select SEG32-SEG13 */
    LCR = 0xFE; /* LCD ON */
    LCR2 = 0xE0; /* A waveform */

    lcdram = LCDRAM; /* Set LCDRAM Address */
    for(i = 0; i <= 0x0F; i++){ /* Initialize LCD RAM */
        lcdram[i] = 0;
    }
}

```

```
}  
}
```

リンクアドレス指定

セクション名	アドレス
CV1	0x0000
CV2	0x001A
CV3	0x001E
CV4	0x0020
P,D	0x0100
B	0xFB80

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.19	—	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。