

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。

H8/300L SLP シリーズ

超音波レンジファインダ (Ultrange)

要旨

本アプリケーションノートでは、H8/38024 SLP MCU を使用した超音波レンジファインダの適用を説明します。MCU により 40kHz の矩形波が生成され、超音波センサを通り送信されます。その反響した超音波は、もう 1 つの超音波センサで受信されます。次に MCU により距離を演算します。その実効範囲は 6cm から 200cm までです。

動作確認デバイス

H8/38024 SLP

目次

1. 論理	2
2. 動作説明	10
3. プログラムコード	16
4. ハードウェアの回路図	26
5. 参考文献	27

1. 論理

1.1 概要

本アプリケーションノートでは、H8/38024F MCU を対象デバイスとして使用します。超音波レンジファインダに使用するソフトウェアは、簡単に移植できるように C 言語で書かれています。

超音波は、可聴音より周波数が高く、ほぼ 20kHz の周波数を越える音です。医療診断や画像に使用する周波数は 10MHz かそれ以上に及びます。高周波であればあるほど波長は短くなり、それにより波長は物質からより敏速に反射 (反響) します。しかし、残念ながら極度に高い周波数は生成と測定が困難です。超音波は、圧電レシーバを使用して検知、測定します。

一般に、超音波は警報装置、モーション探知機、自動車の距離計に使用します。他には、医療診断 (人体の撮像)、洗浄 (油分やよごれの除去)、流量計 (ドップラー効果の利用)、非破壊テスト (材料欠陥の検出)、はんだ付け、などに使用します。

1.2 ソフトウェアの適用

超音波が、超音波センサに反響し戻ってくるのに要した時間を測定することで、距離が計算されます。測定対象物質としては、表面積が大きく、超音波を吸収しないものが理想的です。

本アプリケーションノートでは、H8/38024F CPU ボードを使用します。図 1 に超音波レンジファインダの動作原理を示します。40kHz の超音波を 0.5ms 間送信するのに TMOFH (63 端子) を使用し、反響した波長を検知するのに IRQ0 (72 端子) を使用します。送信後、タイマ C が起動し、タイマカウンタ C (TCC) のカウント数を数え、対象物質の距離を計算します。

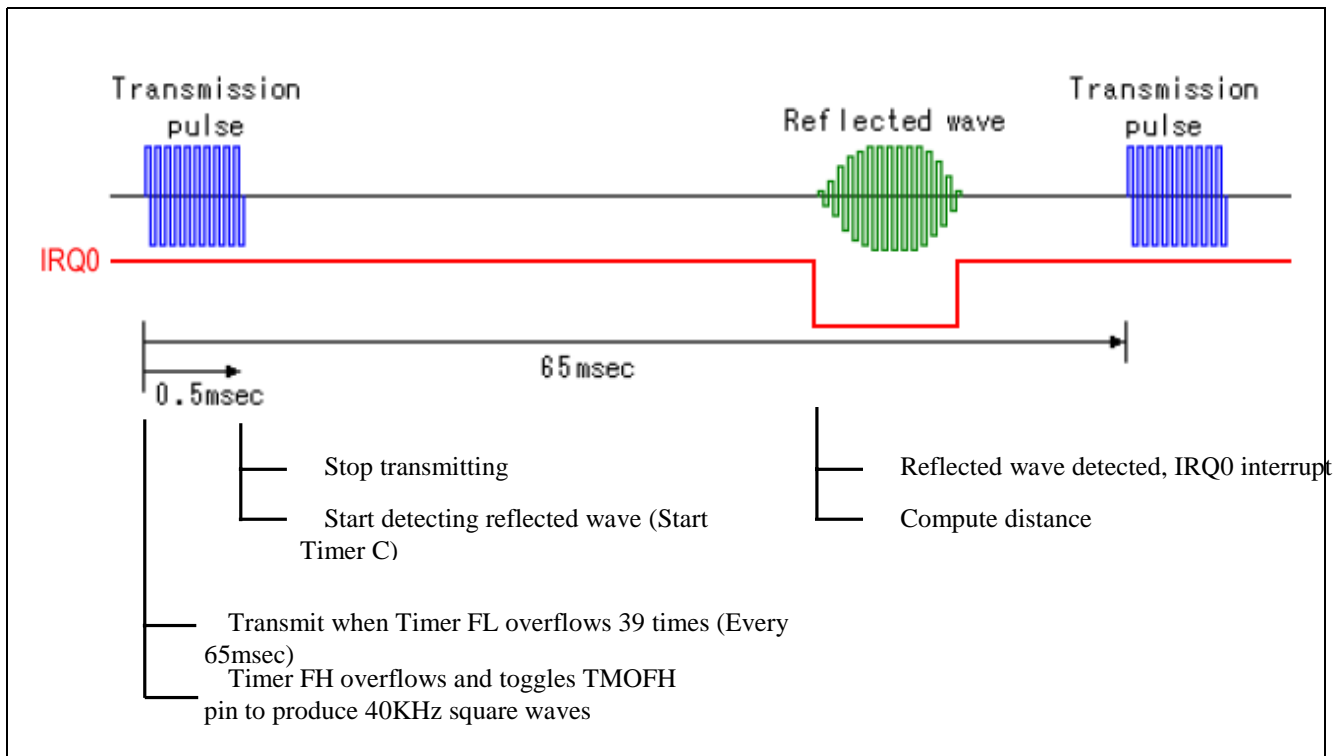


図 1 レンジファインダの動作原理

1.2.1 超音波の送信

タイマ F は 16 ビットのフリーランニングカウンタで、アウトプットコンペアの機能を持ちます。また、2 つの独立した 8 ビットタイマであるタイマ FH とタイマ FL として、それぞれ動作することができます。

本アプリケーションノートでは、タイマ F を 2 つの独立した 8 ビットタイマとして使用します。タイマ FL は、発生するように初期化されます。一方、タイマ FH はコンペアマッチが発生すると、TMOFH 端子にトグル出力します。

表 1 タイマ F のクロックの選択

CKSH2/ CKSL2	CKSH1/ CKSL1	CKSH0/ CKSL0	説明
1	0	0	内部クロック : $\phi/32$ でカウント
1	0	1	内部クロック : $\phi/16$ でカウント
1	1	0	内部クロック : $\phi/4$ でカウント
1	1	1	内部クロック : $\phi_w/4$ でカウント

タイマ FL は $\phi/32$ の内部クロックを選択し、アウトプットコンペアレジスタ FL (OCRFL) には H'FF がロードされます。

したがって、タイマ FL は 1.67ms ごとに割り込みを生成します。以下がその計算方法です。

ϕ = 水晶発振子周波数/2

$$\text{タイマ FL 内部クロック周波数} = \frac{\text{水晶発振子周波数}}{2 \times 32} = \frac{9.8304\text{MHz}}{64} = 153.6\text{kHz}$$

$$\text{割り込み期間} = \frac{1}{153.6\text{kHz}} \times 256 = 1.67\text{ms}$$

65ms ごとに超音波の送信を始めるには、タイマ FL は 39 回 ($65\text{ms}/1.67\text{ms} = 39$) の割り込みを生成する必要があります。

表 2 に、ポートモードレジスタ 3 の設定を示します。P32/TMOFH 端子を I/O 端子として使用するか、TMOFT 出力端子として使用するか、端子を設定します。

表 2 ポートモードレジスタ 3 (PMR3) のビット 2 (TMOFH)

ビット 2 TMOFH	説明
0	P32 I/O 端子として機能 (初期値)
1	TMOFH 出力端子として機能

カウンタ FH (TCFH) の値がアウトプットコンペアレジスタ FH (OCRFH) の値とマッチしたときに、TMOFH 端子にトグル出力するように設定し、タイマ FH が 40kHz の信号を生成します。アウトプットコンペアレジスタ FH の値は、次のように計算します。

タイマ FH の内部クロックとして $\phi/4$ を選択します。

$$\begin{aligned} \text{タイマ FH の内部クロック期間} &= \frac{1}{\frac{\text{水晶発振子周波数}}{2 \times 4}} = \frac{8}{9.8304\text{MHz}} \\ &= 0.814\mu\text{s} \end{aligned}$$

40kHz の信号を生成するには、TMOFH 端子に 12.5 μ s ごとにトグル出力する必要があります：
(1/40kHz)/2。

$$\text{アウトプットコンペアレジスタ FH(OCRFH)} = \frac{12.5\mu\text{s}}{0.814\mu\text{s}} = 15.36 \approx 15$$

よって、OCRFH にロードすべき値は H'0F です。

63 端子を I/O ポートの 32 端子として使用して送信を停止する前に、0.5ms 間、超音波を送信するため、ソフトウェア遅延を使用します。

1.2.2 タイマ C の初期化

超音波の送信後、タイマ C が起動し、反響した超音波が戻り、時間をカウントします。タイマ C をオートリロード機能、 $\phi/64$ 内部クロックアップカウンタに設定します。表 3 にタイマモードレジスタ C の設定を示します。設定内容を太字で示します。

表 3 タイマモードレジスタ C (TMC3) の設定内容

ビット 7 TMC7	説明
0	インターバルタイマ機能選択 (初期値)
1	オートリロード機能選択

ビット 6 TMC6	ビット 5 TMC5	説明
0	0	TCC はアップカウンタ
0	1	TCC はダウンカウンタ
1	* Don't care	UD端子入力によるハードウェア制御 UD端子入力がHighレベル: ダウンカウンタ UD端子入力がLowレベル: アップカウンタ

ビット 2 TMC2	ビット 1 TMC1	ビット 0 TMC0	説明
0	0	0	内部クロック: $\phi/8192$
0	0	1	内部クロック: $\phi/2048$
0	1	0	内部クロック: $\phi/512$
0	1	1	内部クロック: $\phi/64$
1	0	0	内部クロック: $\phi/16$
1	0	1	内部クロック: $\phi/4$ でカウント
1	1	0	内部クロック: $\phi w/4$ でカウント
1	1	1	外部イベント (TMIC) を 立ち上がり/立ち下がりエッジでカウント

タイマロードレジスタ (TLC) に H'00 がロードされ、0 からカウントを開始します。

割り込み許可レジスタ 2 (IENR2) では、IENTC ビット = 1 のとき、タイマ C の割り込みを許可します。タイマカウンタ C (TCC) のカウント値が H'FF に達すると、次のクロック入力でタイマ C がオーバーフローして、割り込みが発生します。タイマ C オーバーフロー割り込みサブルーチンでは、OVERFLOW_COUNT がインクリメントされ、オーバーフロー回数を記録します。

反響した超音波を受信すると、IRQ0 の電圧レベルが低下し、IRQ0 の割り込みが発生します。TMC2 - TMC0 を 1 に設定することで外部クロックを選択します。しかし、実際にはカウンタをインクリメントする外部のクロックがないので、カウンタは停止します。次に TCC の値が読み出され、距離を計算します。

1.2.3 距離の計算

タイマの内部クロックに $\phi/64$ を選択し、距離を次のように計算してください。

$$\begin{aligned} \text{タイマ C は、カウント 1} &= \frac{1}{\frac{\text{水晶発振子周波数}}{2 \times 64}} = \frac{128}{9.8304\text{MHz}} = \\ &13.02\mu\text{s} \end{aligned}$$

$$\text{音の速度} = 343\text{m/s} = 34300\text{cm/s}$$

$$\text{したがって、1cm 移動するのに必要な時間} = 1\text{s}/34300\text{cm} = 29.15\mu\text{s}$$

タイマカウンタ C (TCC) でカウント数とオーバーフロー数を記録することで、対象物質の距離を計算できます。

たとえば、カウント数が 55 でオーバーフローが 1 の場合、

$$\text{カウントの総数} = (1 \times 256) + 55 = 311$$

$$\text{反響した超音波の受信に要した時間 (単位 : } \mu\text{s)} = 311 \times 13 = 4043$$

$$\text{センサと対象物質の間の距離} = \frac{4043/2}{29} = 69.7 \approx 70\text{cm}$$

超音波の反射であるため 2 で割ってください。(移動した距離は対象物質との距離の 2 倍になる。)

1.3 ハードウェアの適用

超音波レンジファインダの概要は、4章で説明します。超音波トランスミッタとレシーバの回路の詳細は、次の章で説明します。

1.3.1 トランスミッタの回路

トランスミッタの回路は、複数のインバータと2つのトランジスタで構成されています。最初のインバータは超音波のマイナスを出力します。次のインバータは並列接続され、送信する電力量を増やします。一方、トランジスタはCMOSインバータを駆動します。センサの位相は、プラスとマイナス端子の間で180°シフトします。トランスミッタに供給される電圧は、1つのインバータから入力される2倍の量です。(プラスとマイナスのピークツーピーク値がある。)

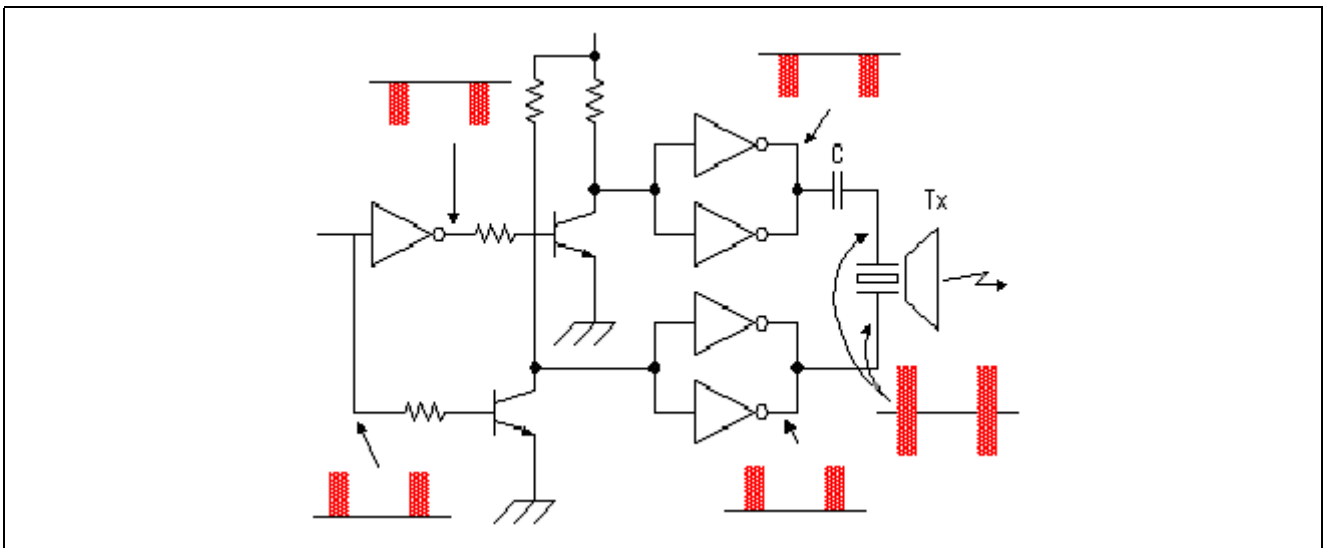


図3 トランスミッタ回路

1.3.2 レシーバの回路

レシーバの回路は、主に信号増幅部と検出回路の2つで構成されています。

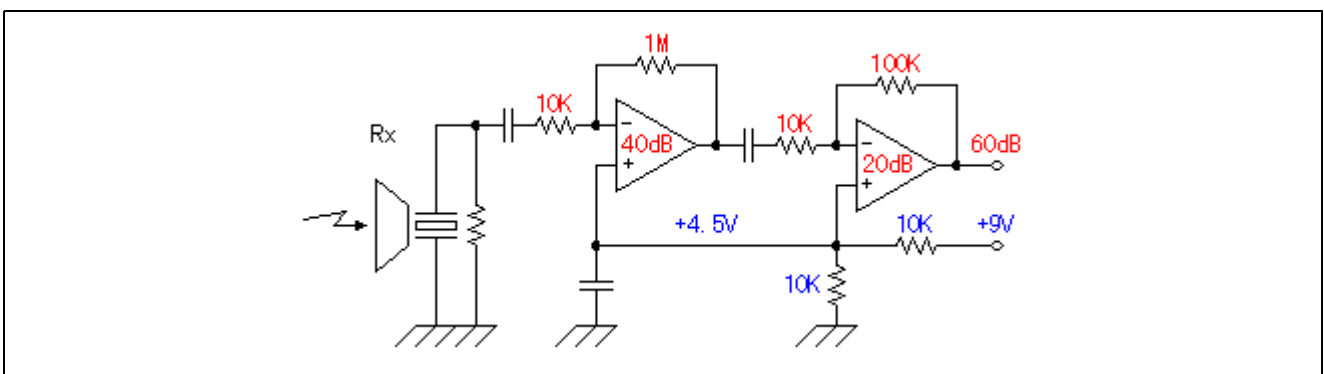


図4 信号増幅回路

レシーバセンサからの超音波信号を受信すると、信号は1000倍に増幅されます。第1段で元の信号が100倍に増幅され(40dB)、第2段の利得は10です(20dB)。

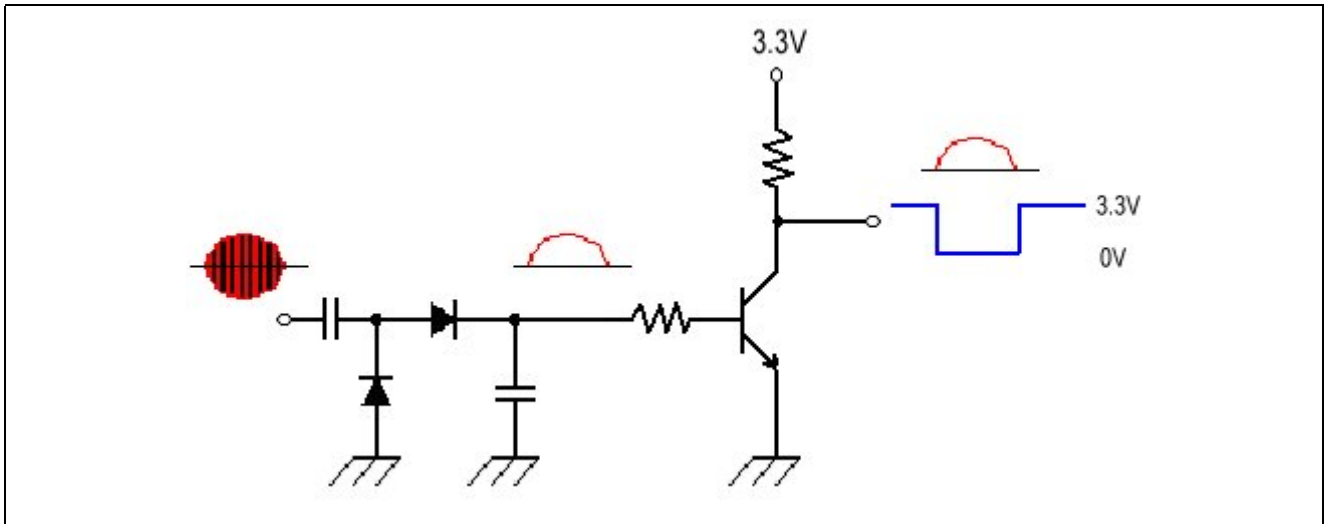


図 5 信号検出回路

増幅回路の次に、信号は半波整流回路から構成される検出回路を通り送信されます。この回路には2つの1CV5ダイオードが実装されており、整流回路を通り送信された信号はトランジスタに送信されます。信号がないときは、出力電圧は3.3V (High)です。信号があるときは、出力電圧は0V (Low)に下がります。

立ち下がりエッジを検出すると、出力が H8/38024 の IRQ0 端子に入力され、割り込みが発生します。

1.3.3 電源

3種類の電圧電源が必要です。

レンジファインダボード

- 9Vの入力電圧：LM833のアンプ用
- 3.3V：74LS04のインバータとBC547のトランジスタ用

38024 CPU ボード

- 5Vの入力電圧：CPUボード
- 3.3V：MCU用

ユーザはレンジファインダボードに9Vの入力電圧を、CPUボードに5Vの入力電圧を供給してください。

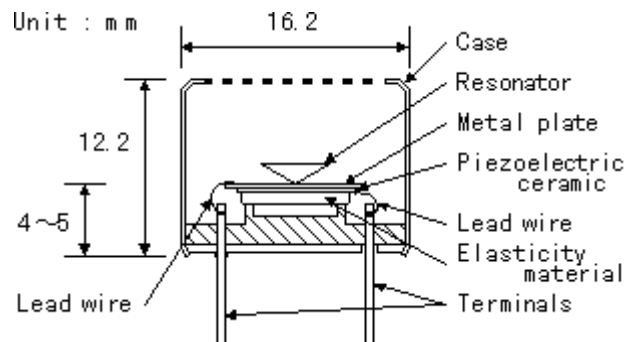
1.3.4 超音波センサ

本アプリケーションノートでは、日本セラミック株式会社製の超音波センサ (T40-16) とレシーバ (R40-16)を使用します。Tはトランスミッタ、Rはレシーバの略です。40は超音波の共振周波数の値 (40kHz)です。



超音波センサの仕様概略を以下に示します。

Item		Spec
Frequency(kHz)		40
Sound pressure level (dB)		115 <
Sensitivity (dB)		-64 <
Size (mm)	Diameter	16.2
	Height	12.2
	Interval	10.0



2. 動作説明

図 6 に示すように、H8/38024F CPU ボードは、超音波レンジファインダの回路に接続されています。

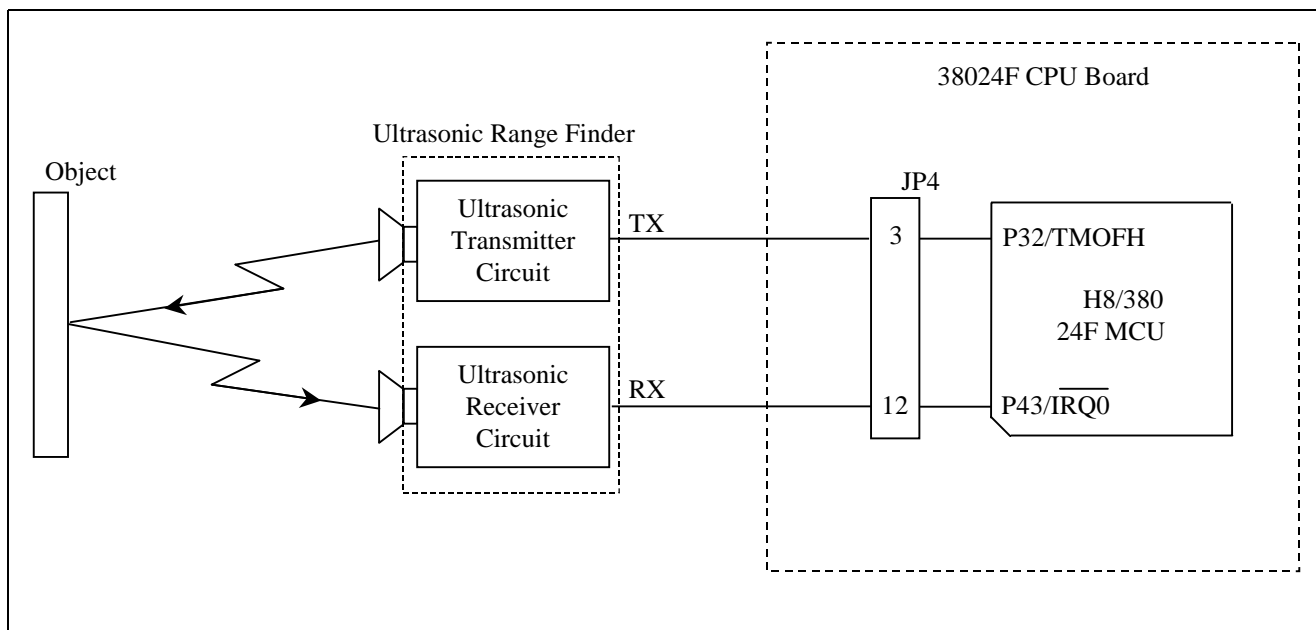


図 6 超音波レンジファインダと MCU の接続

超音波を出力する TMOFH 端子は、超音波トランスミッタ回路の TX 端子に接続してください。検出された信号は IRQ0 端子に接続されているので、H8/38024F CPU ボードの JP4 の 3 端子と 12 端子を、超音波レンジファインダの TX 端子と RX 端子にそれぞれ接続してください。

2.1 ハイパターミナルの設定

ハードウェアの設定が完了したら、ハイパターミナルウィンドウを設定して MCU が検出した距離を表示してください。また COM ポートは、プログラムで使用する UART のプロトコルとボーレートに従い、図 7 に示すように設定してください。

スタートメニューから、Programs Accessories Communications HyperTerminal S の順に検索してください。またハイパターミナルウィンドウの File メニューから Properties を選択し、Configure をクリックし、ポート設定を変更してください。

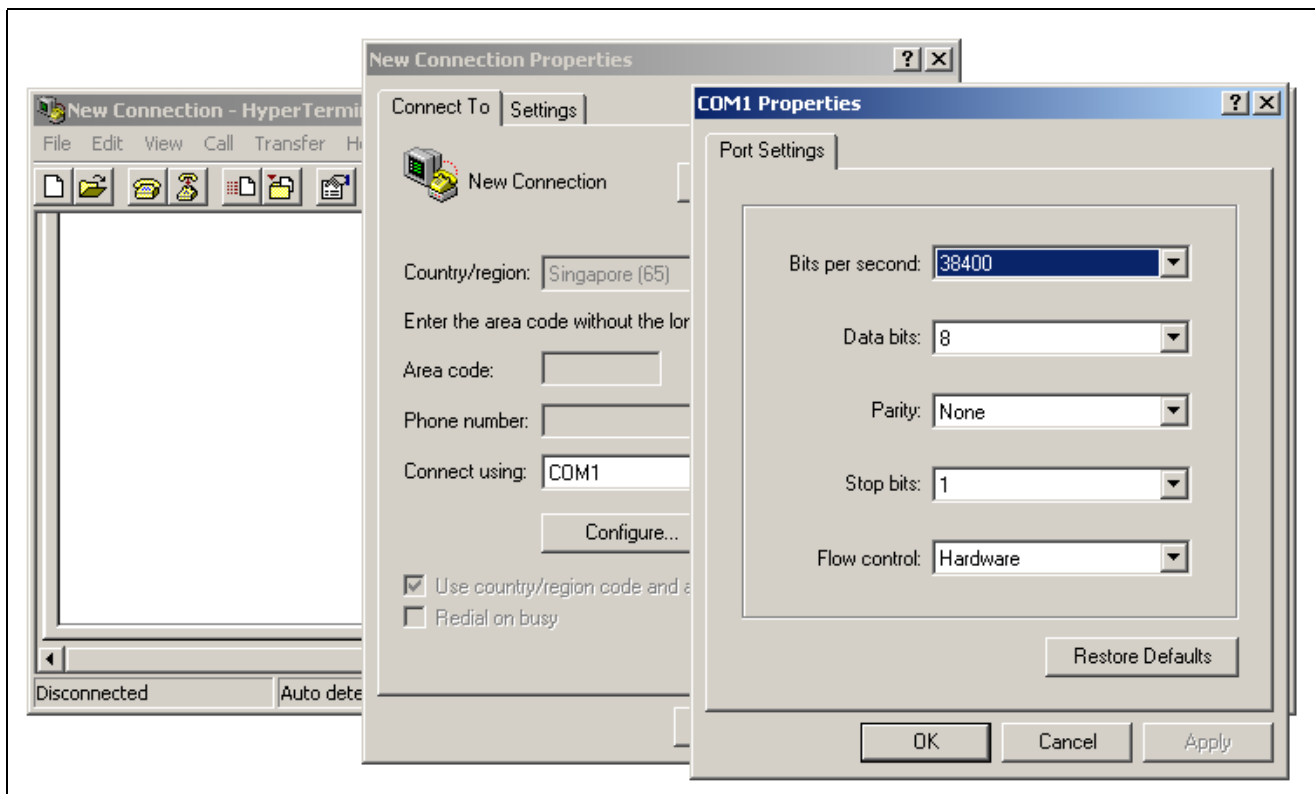


図 7 PC のハイパターミナルの設定

2.2 結果

最初に、FD(フロッピーディスク) からプログラムを MCU にコピーしてください。次に、ユーザモードでリセットボタンを押下するとプログラムが動作します。CPU ボードの LED D1 が継続して点滅していることを確認してください。これは超音波が送信されていることを示します。

センサの前に、超音波を吸収しない大きな物体を設置すると、図9で示すように、ユーザはハイパターミナルウィンドウで、検出された距離を確認することができます。物体が検出される (IRQ0 割り込みが生成される) たびに、ハイパターミナルウィンドウに "." (点) が表示されます。同じような値を 5 回読み出したら、それらの値を平均し、距離を算出し、画面上に表示されます。

この超音波レンジファインダは、6cm から 200cm の距離にある物体を検出することができます。

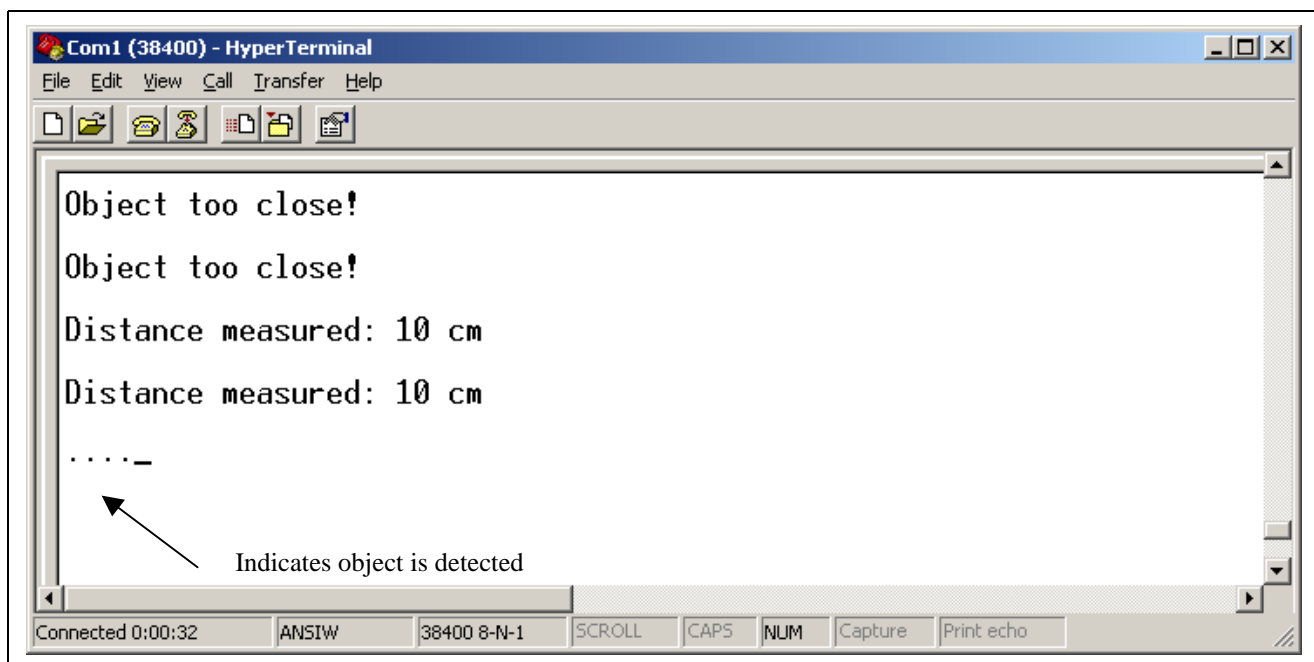


図 8 PC ハイパターミナルに表示された結果

2.3 制限

2.3.1 センサ間の距離

超音波レンジファインダを設計するにあたり、まず考慮しなければならないのは、超音波センサを設置する場所です。超音波センサのレシーバをトランスミッタから遠く離して設置すると、近くにある物体を検出することができません。以下にその図を示します。

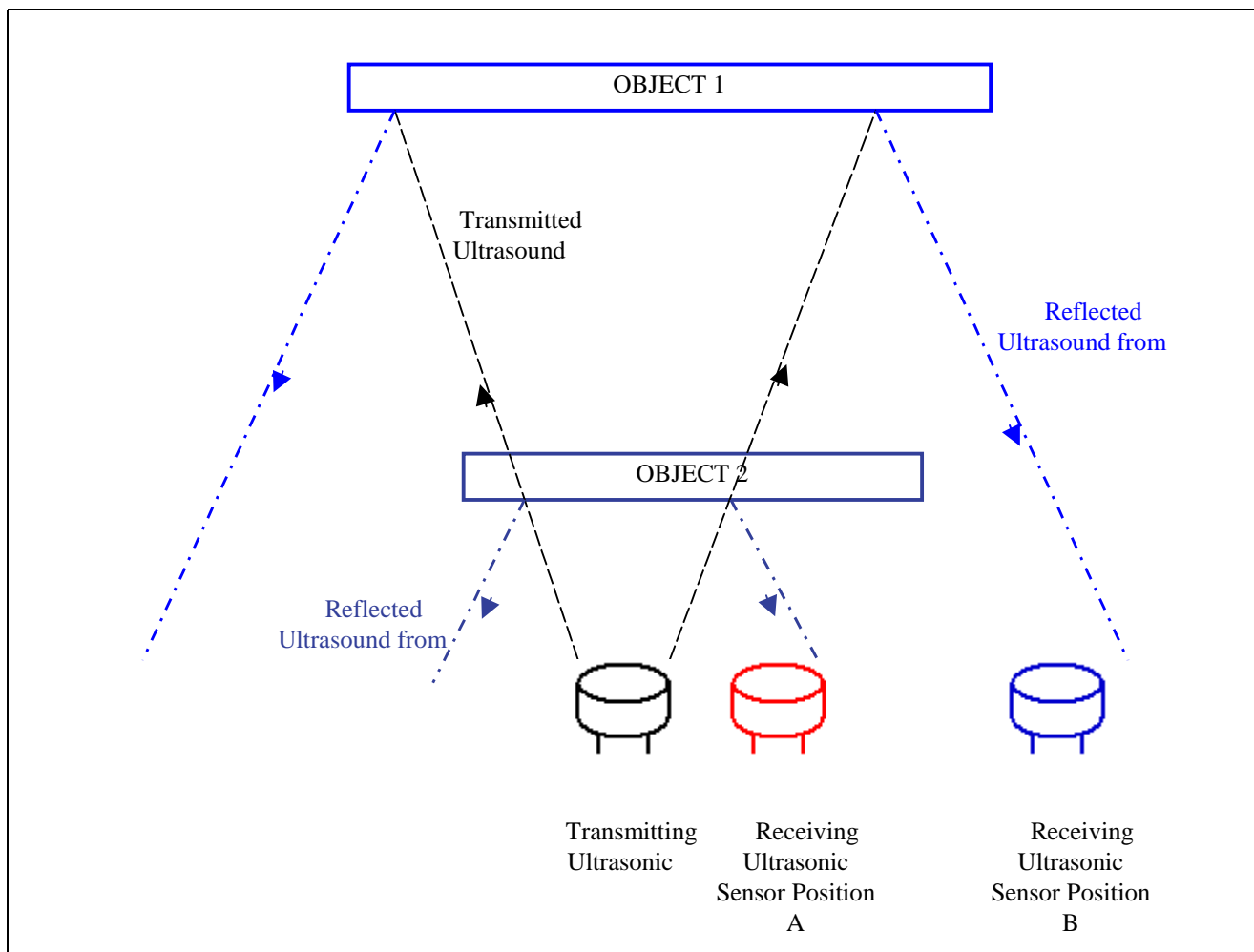


図9 センサの距離間の相違点

遠くに設置した物質 1 の場合、超音波センサのレシーバを位置 A または B のいずれかに設置しても、反射波は両方のセンサに届くので問題ありません。

しかし、物体 2 の場合、超音波センサを位置 B に設置すると、位置 B はトランスミッタと離れているので、反射波を検出しません。

短い距離の測定が必要なマイクロマウスに応用するときは、センサを互いに近くに設置するか、わずかに回転させ、センサが互いに向き合うようにしてください。

本アプリケーションノートでは、センサを 3.5cm 離して設置しています。

2.3.2 実測距離

超音波レンジファインダは、反響した超音波を受信した時間を 1/2 で割ることにより、距離を測定します。しかし、実際の距離は超音波センサに対して垂直な距離です。この誤差は、物質を近くに設置した場合、より顕著ですが、遠くの場合は無視できます。図 10 に実測距離の図を示します。

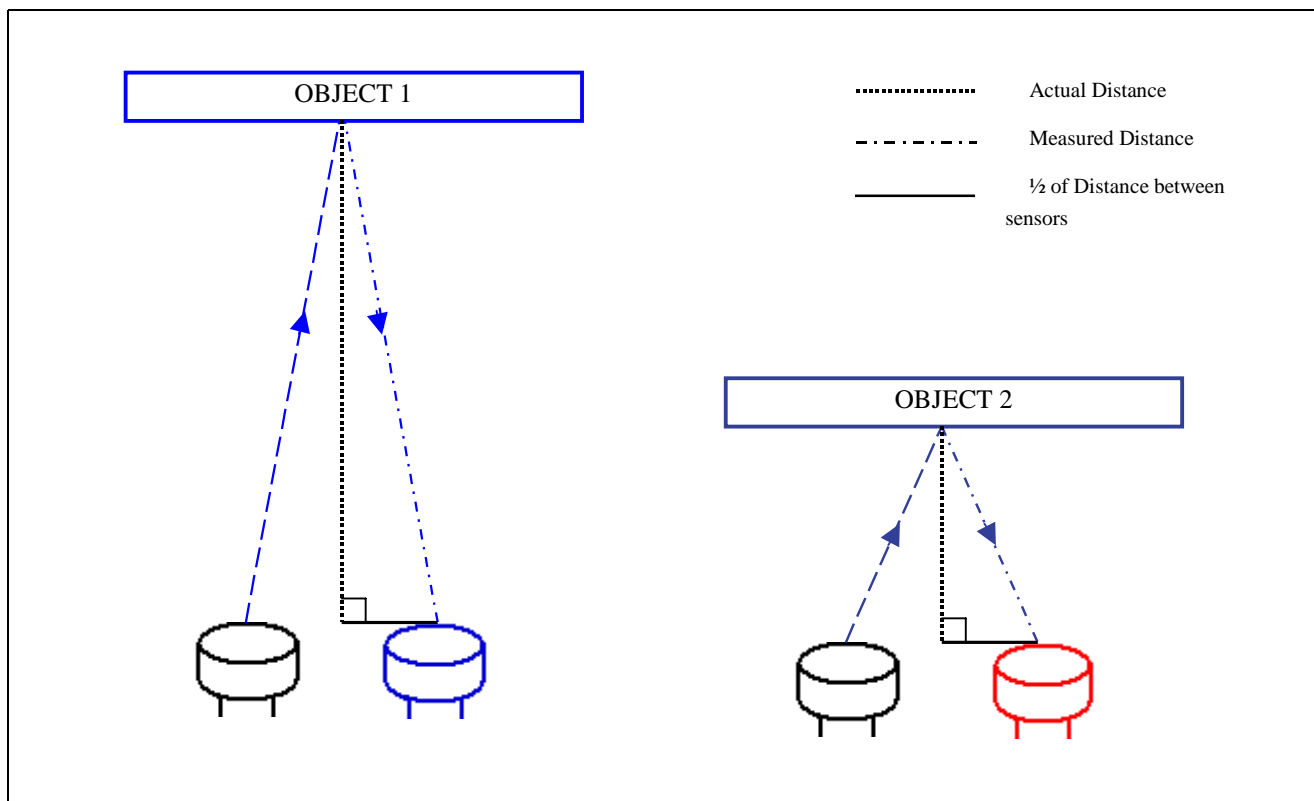


図 10 実測距離の図

ユーザは以下の式で距離を計算し、誤差を修正することができます。

$$\text{実際の距離} = \sqrt{(\text{測定した距離})^2 - (\text{センサ間の距離の}\frac{1}{2})^2}$$

2.3.3 デットゾーン

超音波センサには対象（物）を検出できないデッドゾーンがあります。このデッドゾーンは実際に検知する面と検知できる最小範囲の間の距離です。日本セラミック株式会社製の超音波センサのデッドゾーンは、実験結果により約 1cm です。

2.3.4 検出可能な範囲

検出可能な最小範囲は、回路全体のレスポンスとレイアウトだけではなく、デットゾーン (2.3.3 章を参照) と MCU の条件 (仕様) により決まります。その他の MCU の条件としては、割り込みレイテンシーがあります。距離はタイマ C の値から計算するので、精度はタイマ C のカウント開始と終了の時間に依存します。

検出可能な最小または最大範囲は、実験結果から、それぞれ 6cm と 200cm です。

したがって、プログラムにはオフセット値 (距離 +5) が必要です。ユーザは、実験を行うことで回路の検出可能な距離の最小値を決め、それに応じてオフセットの値を加えてください。

同様に、LM833 のオペアンプへの入力電圧によって、検出可能な最大範囲が決定します。増幅した出力信号の振幅は、入力電圧と共に減少します。したがって、検出可能な最大値も同様に減少します。

LM833 の動作アンプへの最小入力電圧は、このレンジファインダの回路では、+5V です。これは、2 つのダイオードを通過する際の電圧降下によるものです。電圧が +5V より低い場合、トランジスタ Q3 をオンにすることができません。

LM833 のオペアンプへの入力電圧を +5V に下げると、最大検出範囲は 150cm に縮小されます。

3. プログラムコード

以下に添付したプログラムコードは、H8/38024 MCU を対象とした HEW を使用して生成しました。使用するツールチェーンは、HEW のバージョン 2.2 (リリース 15) 用の無償の H8 Tiny/SLP ツールチェーン (バージョン 1.0.0) です。

以下のフローチャートは、ユーザの理解をより深めるために、主要な機能を示しています。

```

/*****
/*
/* FILE      :subfunctions.h
/* DATE      :Tue, Feb 17, 2004
/* DESCRIPTION :Subfunctions and defined constants used
/* CPU TYPE   :H8/38024F
/*
*****/

void initialize (void);
void char_put(char);
void PutStr(char *);
void display_decimal(unsigned int);

#define COUNT_PERIOD 13 // (micro seconds) calculate as (2*64) / XTAL freq
                        // (internal clock source selected as phi/64)
#define CONV_DIST 29 // Speed of sound = 343m/sec,
                     // so for 1cm, time taken is 29.15 usec

/*****
/*
/* FILE      :Ultrasonic_sensor.c
/* DATE      :Tue, Feb 17, 2004
/* DESCRIPTION :Main Program
/* CPU TYPE   :H8/38024F
/*
/* This file is generated by Renesas Project Generator (Ver.2.1).
/*
*****/
#include "iodefine.h"
#include "subfunctions.h"
#include <machine.h>
#include <_h_c_lib.h>

unsigned int TIME, DISTANCE, COUNT, PREVIOUS_COUNT;
unsigned int INPUT_CAPTURE, OVERFLOW_COUNT, MATCH, delay;
unsigned long ADD_COUNT;
signed int DIFFERENCE;

void main(void)
{
    delay = PREVIOUS_COUNT = MATCH = ADD_COUNT = 0;
    initialize();

    while (1);
}

//-----//
    
```

```

/* init_sci() : Initialize PORT 9, SCI3, IRQ0, Timer F          */
//-----//
void initialize (void)
{
// INITIALIZE Port 9
P_IO.PMR9.BYTE = 0x00;          // Port 9 as output port
P_IO.PDR9.BYTE = 0xFF;        // Off LEDs

// INITIALIZE SCI3
//Serial Control Register
//CKE1 = CKE0 = '0': SCK32 functions as I/O port
P_SCI3.SCR3.BYTE &= 0x00; //clear TE & RE

//Serial Mode Register
//SMR : |COM|CHR|PE|PM|STOP|MP|CKS1|CKS0| : |0|0|0|0|0|0|0|0|
//COM : Communication Mode : 0 : asynchronous mode
//CHR : Character Length : 0 : character length = 8 bits
//PE : Parity Enable : 0 : parity bit addition and checking disabled
//PM : Parity Mode : 0 : even parity (no effect since parity is //
// already disabled)
//STOP: Stop Bit Length : 0 : 1 stop bit
//MP : Multiprocessor Mode : 0 : multiprocessor communication function
//
//|CKS1|CKS0| : Clock Select: |0|0| : clock source for baud rate generator
P_SCI3.SMR.BYTE = 0x00;

//Bit Rate Register
//For clk = 10MHz, bit rate = 38400 bps, n = 0, N = 3
P_SCI3.BRR = 3;

//minimum of 1-bit delay = 417ns
nop();
nop();
nop();

//SPCR : |---|---|SPC32|---|SCINV3|SCINV2|---|---| : |1|1|1|0|0|0|0|0|
//SPC32 = 1 : P42 functions as TXD32 output pin
//need to set TE bit in SCR3 after setting this bit to 1
//SCINV3 = 0 : TXD32 output data is not inverted
//SCINV2= 0 : RXD32 input data is not inverted
//Bits 7 and 6 are reserved and always read as 1
//Bits 4, 1 and 0 are reserved and only 0 can be written to these bits
P_SCI3.SPCR.BYTE = 0xE0;

P_SCI3.SCR3.BYTE |= 0x30; //Set TE & RE

// INITIALIZE IRQ0 INTERRUPT
P_IO.PMR2.BIT.IRQ0 = 1;          // I/O pin used as input capture
P_SYSCR.IEGR.BIT.IEG0 = 0; // Interrupt at falling edge of IRQ0
P_SYSCR.IENR1.BIT.IEN0 = 1;    // Enable IRQ0 interrupt

// INITIALIZE TIMER F
//Timer Control Register F
//TOLH = '1': Initial output for TMOFH is high
//CKSH2 = '1', CKSH1 = '1', CKSH0 = '0': 8-bit mode, phi/4 i.e., //(9.8304MHz/2/4 =
1.2288MHz)
//TOLL = '0': Initial output for TMOFL is low
    
```

```

//CKSL2 = '1', CKSL1 = '0', CKSL0 = '0': 8-bit mode, phi/32 i.e., //(9.8304MHz/2/32
= 153.6kHz)
    P_TMRF.TCRF.BYTE = 0xE4;

//Timer Control/Status Register F
//CCLRH = '1': in 8-bit mode, TCFH clearing by compare match is enabled
//CCLRL = '1': in 8-bit mode, TCFH clearing by compare match is enabled
    P_TMRF.TCSR.BYTE = 0x11;

//Output Compare Register FL
//OCRF = FF
    P_TMRF.OCRF.BYTE.L = 0xFF;
//Output Compare Register FH
//OCRF = 0F (1.2288MHz/40kHz=30 , 30/2=15 0xF)
    P_TMRF.OCRF.BYTE.H = 0x0F;

    P_SYSCR.IENR2.BIT.IENTFL = 1;// Enable Timer F L interrupt
}
//
//-----//
/* char_put() : Transmits a character to the PC for debugging purposes. */
//-----//

void char_put(char OutputChar)          //Serial Port
{
    //TDRE : transmit data register empty
    while ((P_SCI3.SSR.BIT.TDRE) == 0);          //Wait for TDRE = 1

    P_SCI3.TDR = OutputChar;
    //TEND : transmit end
    while ((P_SCI3.SSR.BIT.TEND) == 0);          //Wait for TEND = 1

    P_SCI3.SSR.BIT.TEND = 0;
}
//
//-----//
/* PutStr() : Transmits a string of characters to the PC */
//-----//
void PutStr(char *str)
{
    while (*str != 0) char_put(*str++);
}
//
//-----//
/* display_decimal() : Transmit through SCI3 an interger value in decimal*/
//-----//
void display_decimal(unsigned int display_data)
{
    unsigned char    first_digit, second_digit, third_digit, fourth_digit, fifth_digit;

    first_digit = (unsigned char)(display_data / 10000);
    second_digit = (unsigned char)((display_data - first_digit * 10000) / 1000);
}

```

```
    third_digit = (unsigned char) (((display_data - first_digit * 10000) - second_digit
* 1000) / 100);
    fourth_digit = (unsigned char) (((display_data - first_digit * 10000) - second_digit
* 1000) - third_digit * 100) / 10);
    fifth_digit = (unsigned char) (((display_data - first_digit * 10000) - second_digit
* 1000) - third_digit * 100) - fourth_digit * 10);

    if (display_data >= 10000) char_put(first_digit + '0');
    if (display_data >= 1000) char_put(second_digit + '0');
    if (display_data >= 100) char_put(third_digit + '0');
    if (display_data >= 10) char_put(fourth_digit + '0');
    char_put(fifth_digit + '0');
}
// _____
```

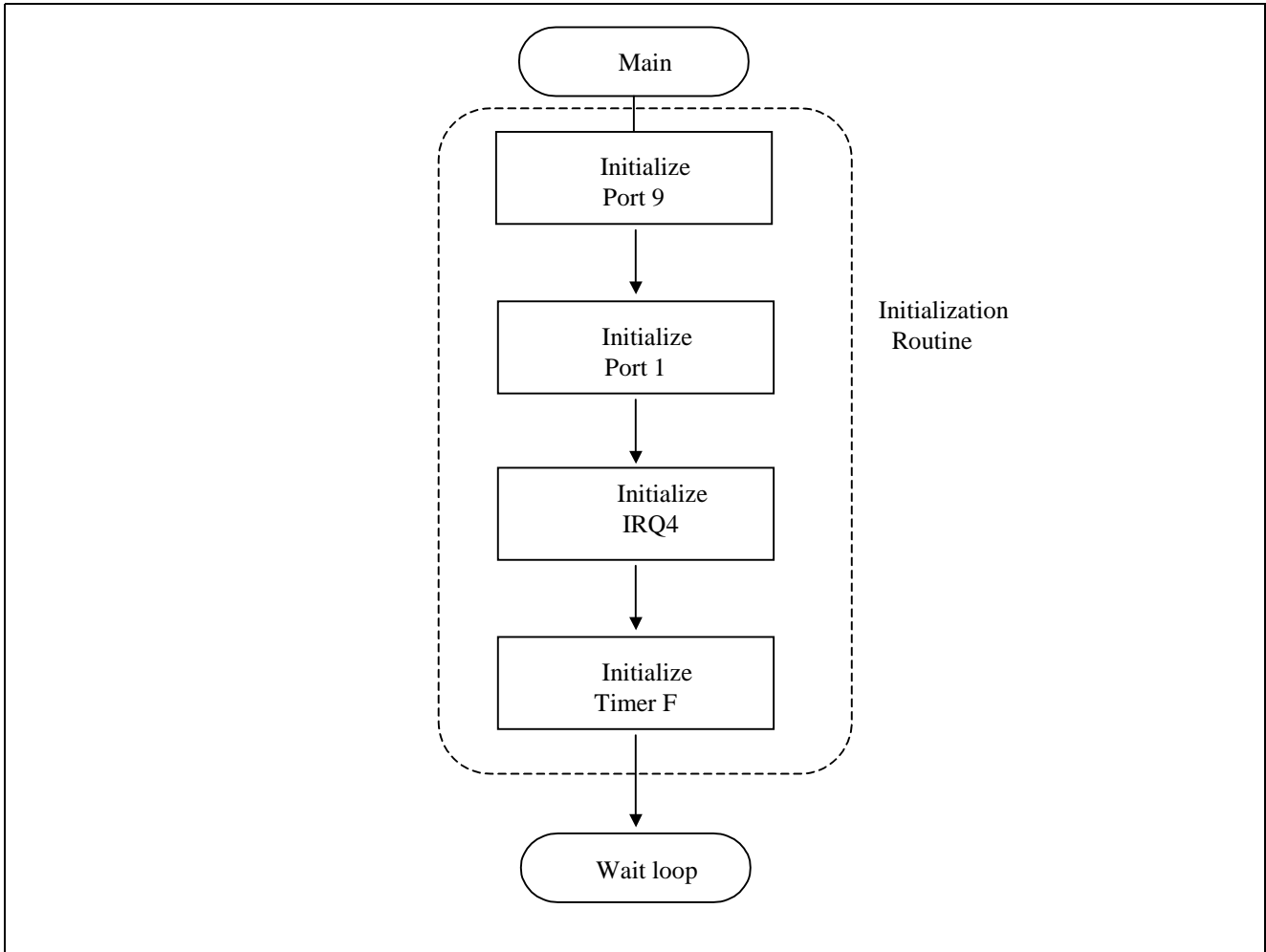


図 9 メインプログラム

```

/*****/
/*                                     */
/* FILE      :intprg.c                */
/* DATE      :Tue, Feb 17, 2004       */
/* DESCRIPTION :Interrupt Program     */
/* CPU TYPE   :H8/38024F              */
/*                                     */
/* This file is generated by Renesas Project Generator (Ver.2.1). */
/*                                     */
/*****/
#include "iodefine.h"
#include "subfunctions.h"
#include <machine.h>

extern unsigned int TIME, DISTANCE, COUNT, PREVIOUS_COUNT;
extern unsigned int INPUT_CAPTURE, OVERFLOW_COUNT, MATCH, delay;
extern unsigned long ADD_COUNT;
extern signed int DIFFERENCE;

#pragma section IntPRG

// Vector 4 IRQ0
__interrupt(vect=4) void INT_IRQ0(void)
{
    int a;

    COUNT = TIME = DISTANCE = 0;

//STOP timer C counting as tigger selected as external source
    P_TMRC.TMC.BYTE = 0x9F;           // Auto reload, External Trigger (TMIC)

    INPUT_CAPTURE = (unsigned char) P_TMRC.TCCTLC; // Read Timer C counter

    COUNT = (INPUT_CAPTURE + (OVERFLOW_COUNT * 256));
    PutStr(".");

    for (a = 0; a < 300; a ++);      // Short delay

    DIFFERENCE = COUNT - PREVIOUS_COUNT;

    if (DIFFERENCE > -4 && DIFFERENCE < 4)
    {
        MATCH ++;
        ADD_COUNT += COUNT;
    }

    PREVIOUS_COUNT = COUNT;

    if (MATCH == 5) // Only display the distance detected after getting 5 //similar readings
    {
        TIME = (ADD_COUNT/5)* COUNT_PERIOD;

        DISTANCE = TIME / (CONV_DIST*2);

        if (DISTANCE == 0)
            PutStr("¥rObject too close!¥r¥n¥n");
    }
}

```

```

        else
        {
            PutStr("Distance measured: ");
            display_decimal(DISTANCE + 5); // Offset of 6cm required
            PutStr(" cm\r\n");
        }

        ADD_COUNT = 0;
        MATCH = 0;
    }

    OVERFLOW_COUNT = 0; // Clear overflow counter
    P_SYSCR.IRR1.BIT.IRRIO = 0; // Clear interrupt request
}

// Vector 13 Timer C Overflow
__interrupt(vect=13) void INT_TimerC(void)
{
    OVERFLOW_COUNT++; // Increment overflow counter when Timer C overflows
    P_SYSCR.IRR2.BIT.IRRTC = 0; // Clear interrupt request
}

// Vector 14 Timer FL Overflow
__interrupt(vect=14) void INT_TimerFL(void)
{
    unsigned int i;

    P_TMRF.TCSRFB.BIT.CMFL = 0; // Clear overflow flag OVFL

    delay++;
    if (delay == 39) // Transmit waveform every 65ms (1/153.6kHz) * 256 * 39
    {
        delay = 0;
        OVERFLOW_COUNT = 0;
        P_IO.PDR9.BIT.P92 ^= 1; //Toggle LED D1 on P92

        // TRANSMIT
        P_IO.PMR3.BIT.TMOFH = 1; // P32 functions as TMOFH to output 40kHz
        for (i = 0; i < 312; i ++); // 0.5ms delay (transmit pulses for 0.5ms)
        P_IO.PMR3.BIT.TMOFH = 0; // P32 functions as I/O pin, pulses stopped
        // INITIALIZE TIMER C TO START DETECTION
        P_IO.PMRB.BIT.IRQ1 = 0; // PB3 functions as IO pin
        P_TMRC.TMC.BYTE = 0x9B; // Auto Reload, up-counter & internal //clk= phi/64
        P_TMRC.TCCTLC = 0x00; // Set counting from 0 (TLC=0)
        P_SYSCR.IENR2.BIT.IENTC = 1; // Enable Timer C interrupt
    }
    P_SYSCR.IRR2.BIT.IRRTFI = 0; // Clear interrupt request
}

```

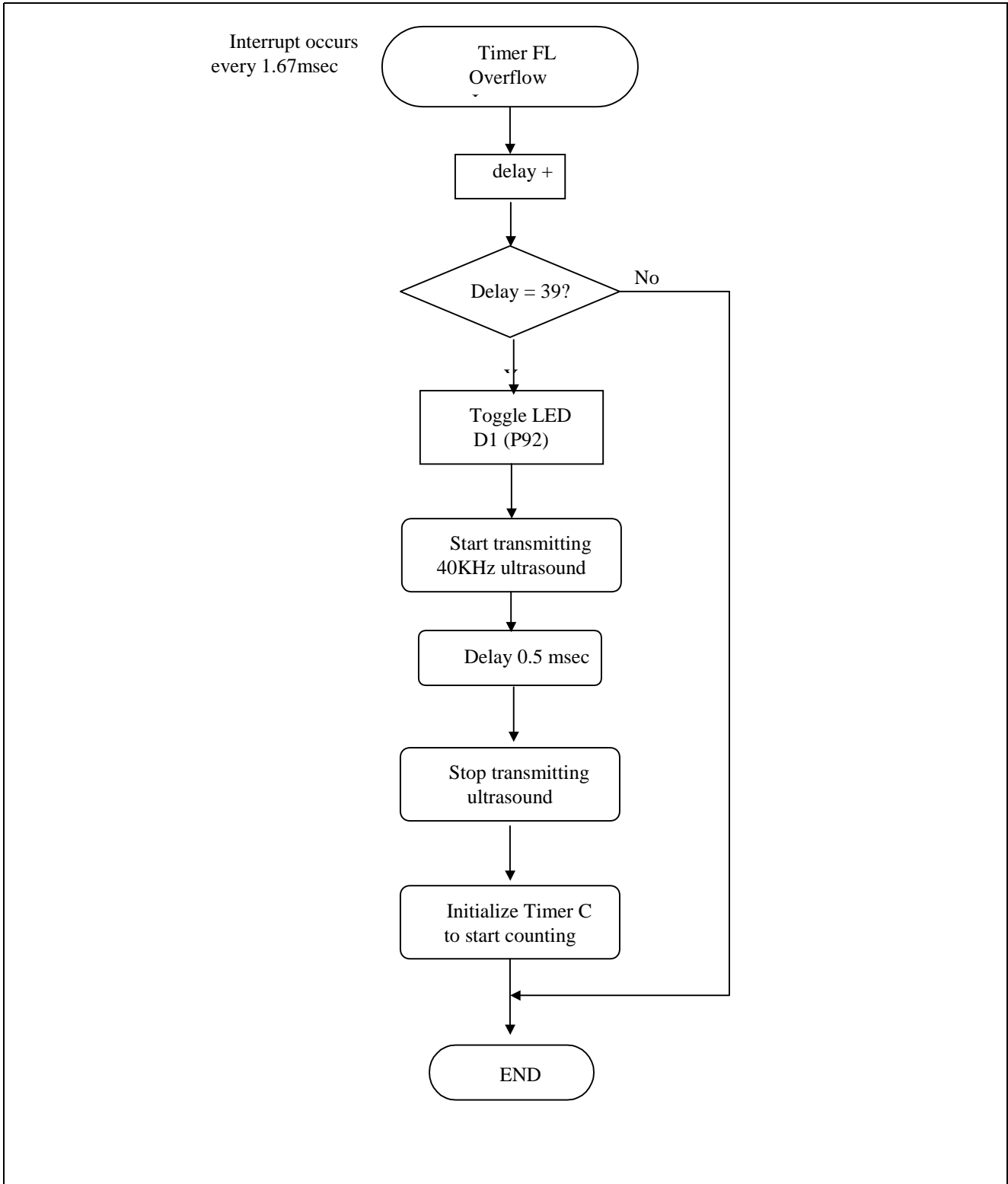



図 10 タイマ FL のオーバーフロー割り込みサービスルーチン

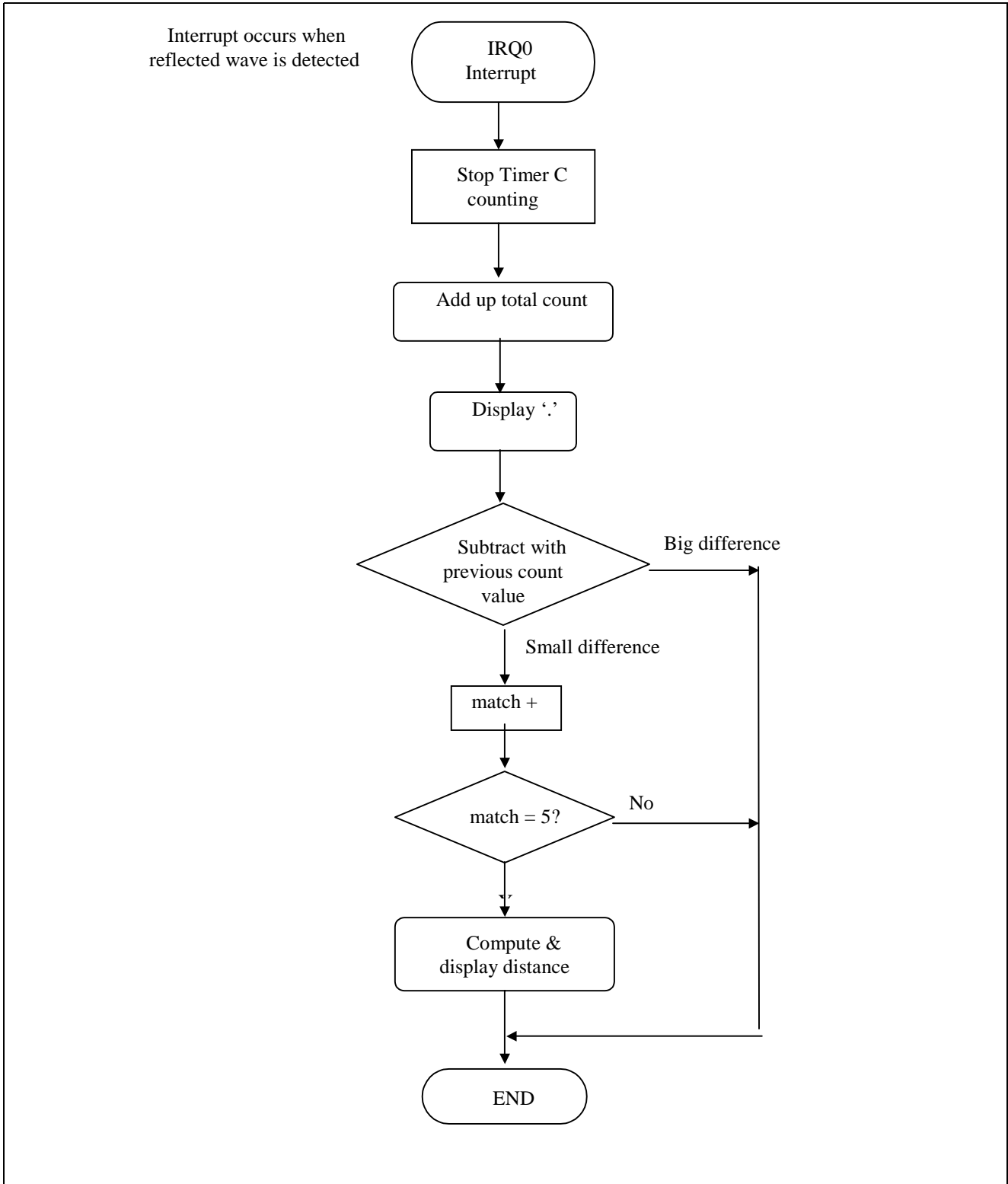


図 11 IRQ0 端子の割り込みサービスルーチン

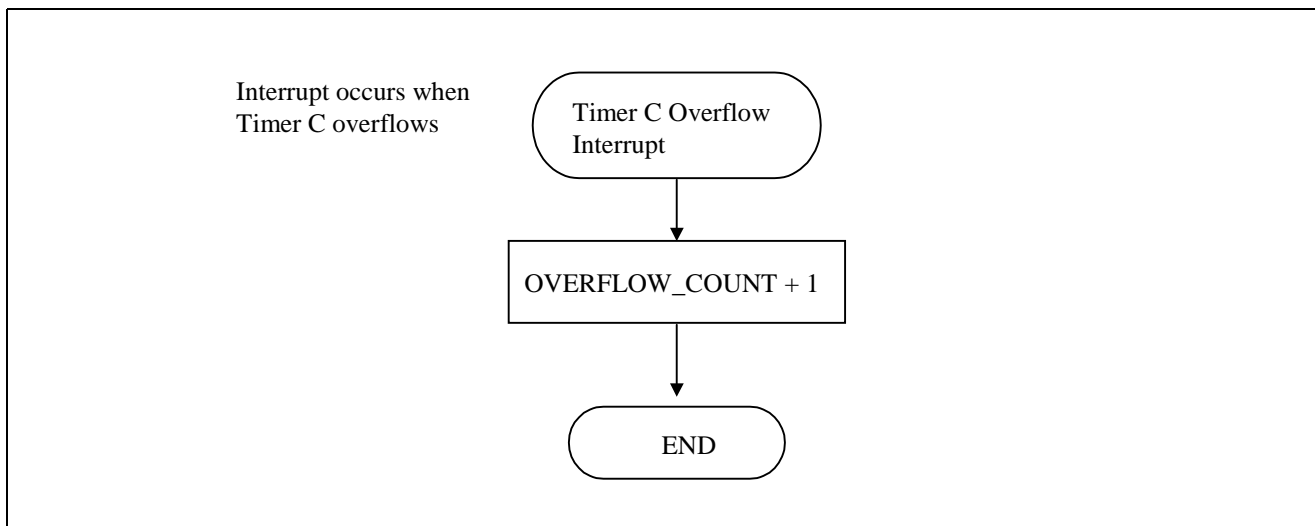
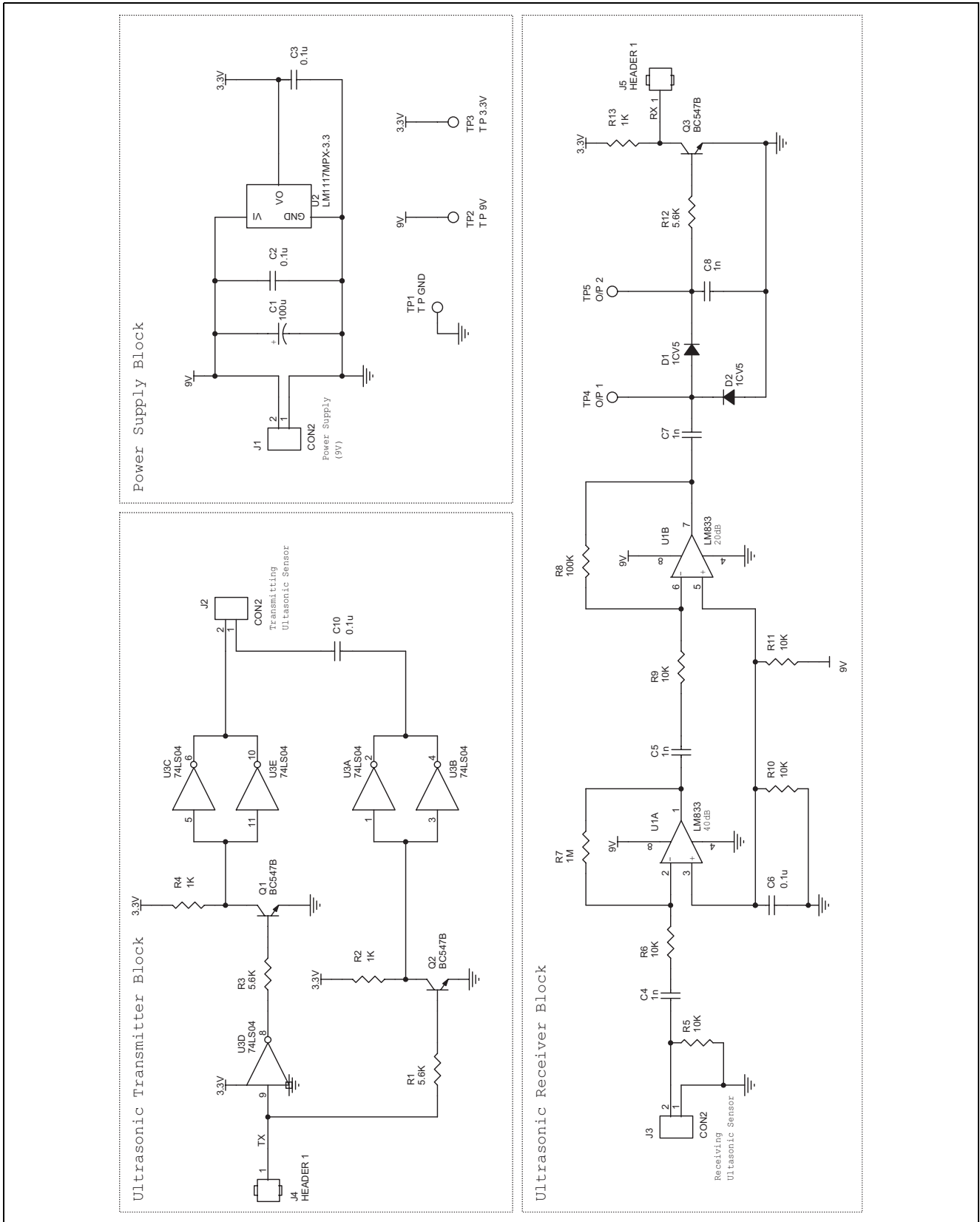


図 12 タイマ C のオーバーフロー割り込みサービスルーチン

4. ハードウェアの回路図



5. 参考文献

1. Open aperture Type - Air Transmission Ultrasonic Sensor, NIPPON CERAMIC CO., LTD.
2. LM1117/LM1117I 800mA Low-Dropout Linear Regulator, 2002, National Semiconductor Corporation.
3. LM833 Dual Audio Operational Amplifier, 2002, National Semiconductor Corporation.

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.08.06	—	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すは、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、その他お気付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。