

ユニークな発想でPICマイコンを使いこなそう

## PIC活用アイデア集

秦 明宏

PICマイコンに関する情報はWeb・サイトや雑誌などでたくさん見かけるようになりました。またPICマイコンに関する入門書も多く出版されています。入門者は容易にそれらの製作記事にしたがって、PICマイコンの工作品を作り上げることができるでしょう。しかし、独自の発想で作品を作り上げようとすると、さまざまな壁にぶつかります。

たとえば「多くの機能を限られたピン数で実現したい」「ある機能を、限られた部品により低コストで実現したい」などです。そのようなとき本書で解説してきたハードウェアの知識があれば、解決のためのユニークなアイデアが浮かぶはずで、本章ではそのアイデアの一部を紹介していきます。

## 5-1 1個の入力ポートでトリマ位置を読み込む

シュミット・トリガ・インバータ1個と抵抗、コンデンサ各1個でトリマの位置を読み込む回路を紹介します。この回路は入力ポート1個で実現しています。

## ● 動作説明

図5-1(a)が回路図です。この回路のシュミット・トリガ・インバータの周辺回路はパルス発振回路になっています。シュミット・トリガ・インバータの出力をCR回路を通して入力に戻すと、入力レベルがしきい値に達したときに出力が反転します。シュミット・トリガ・インバータの入力は二つのしきい値をもっているため入力(A点)の波形は図5-1(b)のようにCRの時定数で二つのしきい値間を往復します。また出力(B点)にはパルスが発生します。

出力波形(B点)の周期は下の式で表されます。

$$\begin{aligned}
 tw_1 &= -\ln \frac{V_{DD} - V_{TH+}}{V_{DD} - V_{TH-}} \times T \\
 tw_0 &= -\ln \frac{V_{TH+}}{V_{TH-}} \times T \dots\dots\dots (5-1)
 \end{aligned}$$

ここで $T = CR$ です。図5-1(a)の回路の場合 $C = C_1$ 、 $R = R_1 + VR_1$ となります。この式からわかるように、

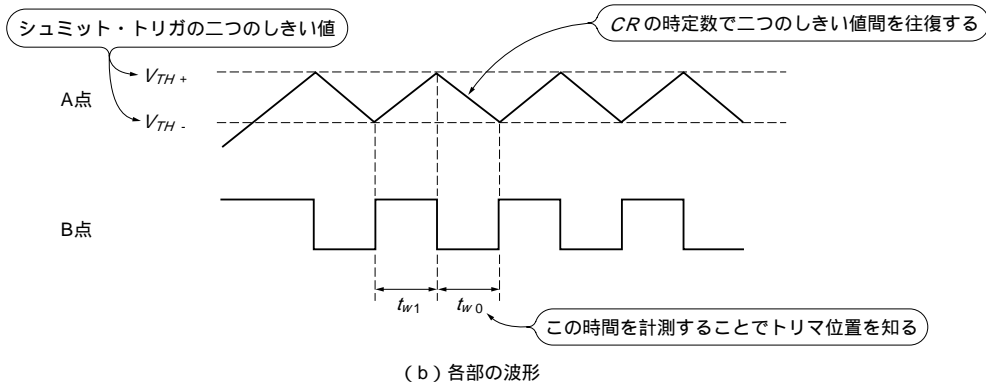
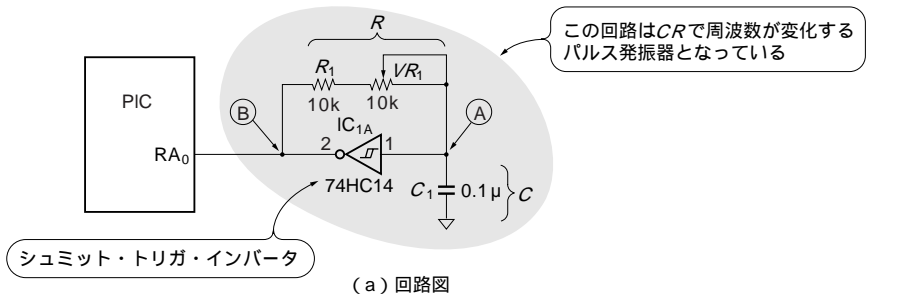


図5-1 シュミット・トリガ・インバータによるトリマ位置読み込み

出力波形の周期はCRに比例します。つまり、トリマを回すとその回転角に応じて周期も変わります。したがって、この周期を計測することでトリマの位置を知ることができます。

図5-1(a)の回路でトリマがmaxとminのときの $t_{w0}$ の実測値は以下のようでした。

条件  $V_{DD} = 5.0V$

< 実測値 >

$V_{TH+} = 3.0V$

$V_{TH-} = 1.8V$

トリマmaxのとき ( $R = 20k$ )

$t_{w0} = 1.0ms$

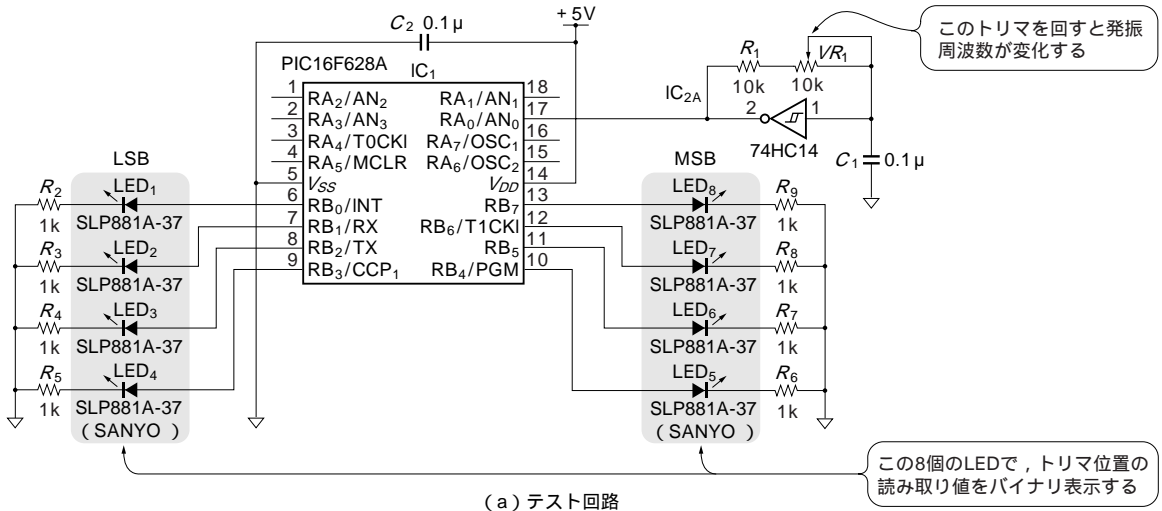
トリマminのとき ( $R = 10k$ )

$t_{w0} = 0.5ms$

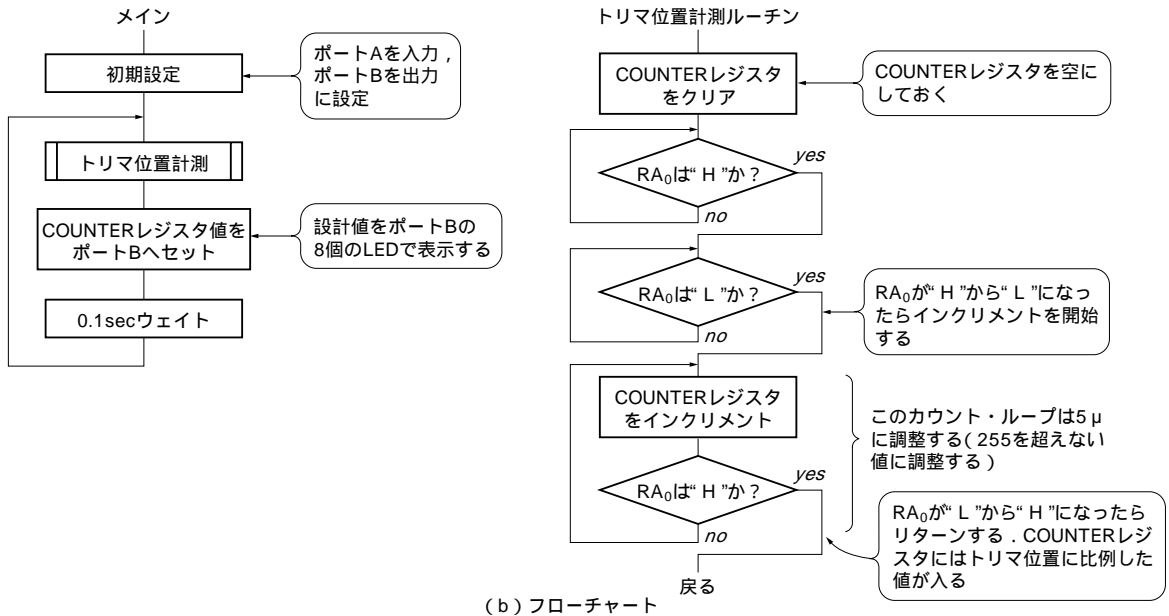
## ● プログラム

図5-2はこの方法を使用してトリマ値を8ビット・バイナリ値で読み取り、その結果をポートBに表示させる回路とプログラムのフローチャートです。プログラムをリスト5-1(稿末)に示します。回路ではポートBのすべてのビットにLEDを付けてあり、このLEDの点灯でバイナリ値を読み取ります。また、プログラムでは初期設定においてRA<sub>0</sub>を入力、RB<sub>0</sub>からRB<sub>7</sub>を出力に設定します。

次に「トリマ位置の計測」、「計測値をポートBに出力」、「0.1秒のウェイト」という動作でループします。



(a) テスト回路



(b) フローチャート

図5-2 トリマ位置読み取り回路

トリマ位置計測ルーチンではRA<sub>0</sub>が“L”レベルの間、基準時間でループしそのループ回数をカウントします。このカウント値がトリマ位置に対応した計測値です。このプログラムを実行すると0.1秒周期でトリマ位置を計測しその結果を8個のLEDに表示します。

なお、 $t_{w0}$ つまりRA<sub>0</sub>が“L”レベルの時間を計測する基準周期は $t_{w0}$ が実測値のmaxである1.0msの場合でも、計測値が255を超えないように5 $\mu$ sにしてあります。したがって、計測値は以下の式で表されます。

$$\text{計測値} = t_{w0}(\text{ms}) / 0.005(\text{ms})$$

ここでトリマがminからmaxまでの $t_{w0}$ の変化は0.5から1.0msなので、計測値は100から200になります。またこれをバイナリで表すと「0110 0100」から「1100 1000」になります。

## 5-2 2個出力ポートで3個のLEDを点灯させる

1個のORゲートと1個のショットキ・バリア・ダイオードを使用し、2個の出力ポートで3個のLEDを点灯させる方法を紹介します。なお、この方法では全部消灯またはどれか一つの点灯という動作となり、2個以上を同時に点灯させることはできません。

### ● 動作説明

図5-3が回路図と動作表です。この図でRA<sub>0</sub>とRA<sub>1</sub>の両方が“H”レベルのときにはいずれのLEDにも電流が流れないため全部消灯となります。また、RA<sub>0</sub>とRA<sub>1</sub>のいずれかが“L”レベルの場合、LED<sub>1</sub>またはLED<sub>2</sub>に電流が流れ点灯します。このときORゲートの出力は“H”レベルなのでLED<sub>3</sub>には電流は流れません。

次に、RA<sub>0</sub>とRA<sub>1</sub>の両方が“L”レベルの場合、ORゲートの出力が“L”レベルになるのでLED<sub>3</sub>に電流が流れ、点灯します。この場合LED<sub>1</sub>とLED<sub>2</sub>にも電流が流れそうですが、図5-3(a)のようにショットキ・バリア・ダイオードにも電流が流れるため、A点の電位はショットキ・バリア・ダイオードの順方向電圧 = 0.4Vになります。

この状態では十分な電流をLED<sub>1</sub>、LED<sub>2</sub>に流すことができず、点灯しません。

### ● プログラム

図5-4はテスト・プログラムのフローチャートです。約1秒間隔でCOUNTERレジスタの値をインクリメントしながらその値をポートAにセットします。したがって、RA<sub>0</sub>とRA<sub>1</sub>の出力値は4通りに切り替わっていきます。

これにより、全部消灯 LED<sub>1</sub>点灯 LED<sub>2</sub>点灯 LED<sub>3</sub>点灯を繰り返します。プログラムをリスト5-2(稿末)に示します。

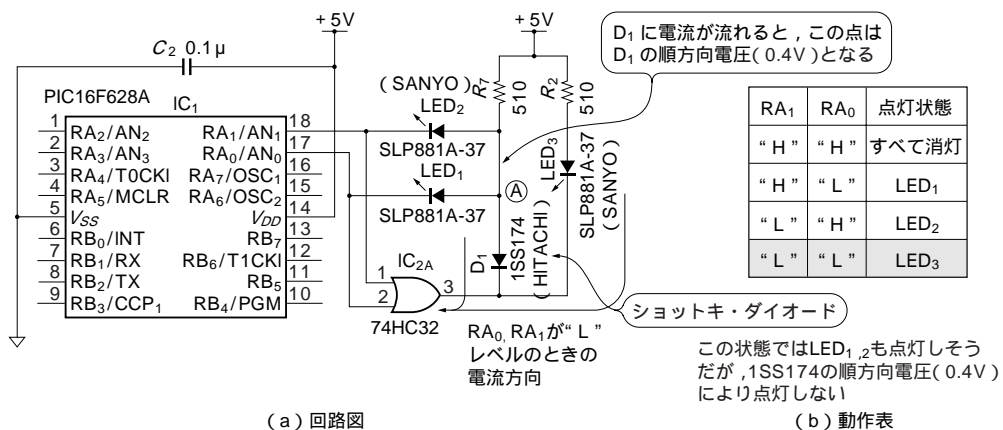


図5-3 2個の出力ポートで3個のLEDを点灯

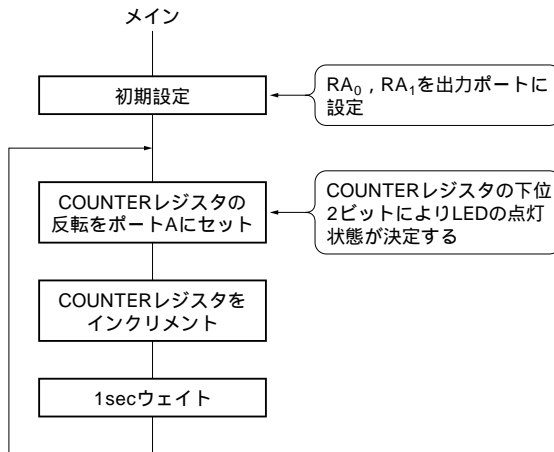


図5-4 3個のLED順次点灯のフローチャート

### 5-3 2個の出力ポートで8個のLEDを点灯させる

シフト・レジスタ(74HC164)を使用し、2個の出力ポートで8個のLEDを点灯させる回路を紹介します。この方法では8個のLEDに任意の点灯をさせることが可能です。ただし、点灯を切り替えるときには常に8個のLEDデータをセットし直す必要があります。

#### ● 動作説明

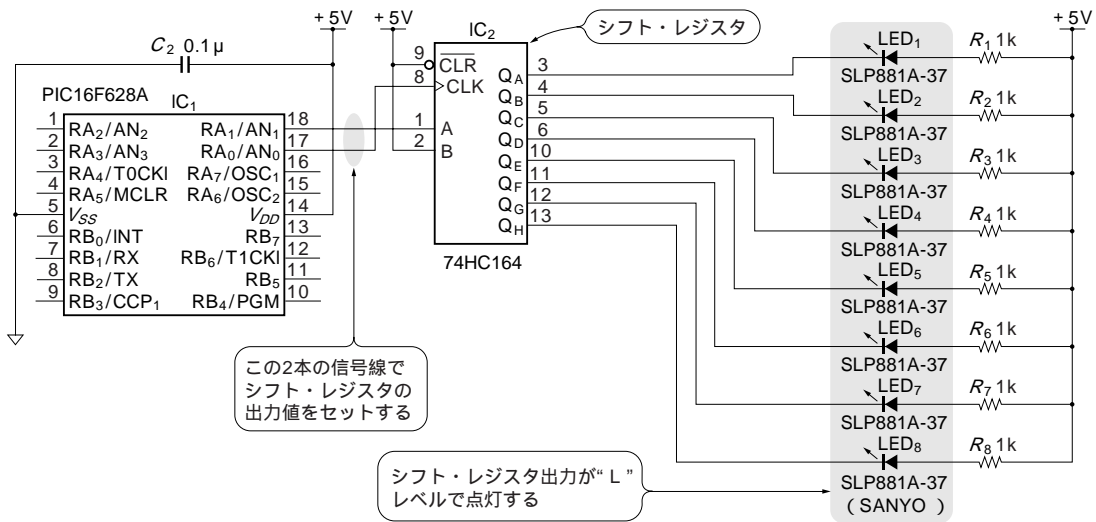
シフト・レジスタ(74HC164)は、CLK入力が立ち上がるたびにAまたはBの入力レベルを $Q_A$ に出力すると同時に、各出力値を上位ビットにシフトして行きます。図5-5(b)のタイムチャートのように、この動作を8回繰り返すことで、8ビットの出力レベルをセットすることができます。

LEDの点灯状態を変更するときには、たとえ1個のみの変更であってもこの動作を行う必要があります。図5-5(a)が回路図です。74HC164のB端子とCLR端子は“H”レベル(5V)に固定しておきます。この設定によりCLKの立ち上がりでA端子入力が読み込まれます。

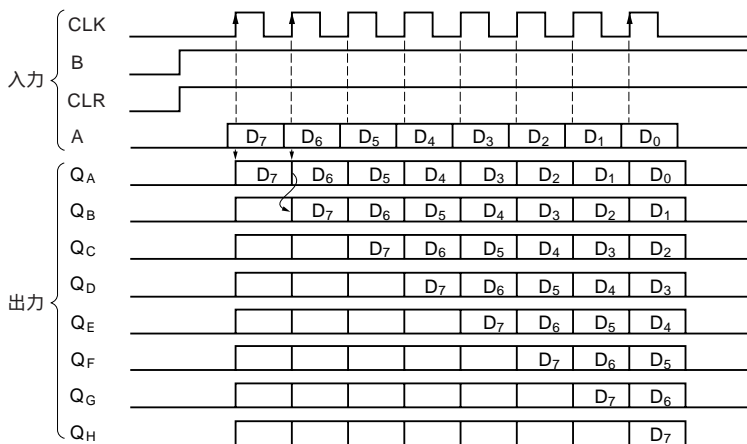
#### ● プログラム

図5-6はテスト・プログラムのフローチャートです。このプログラムでは8ビット・カウンタを約0.2秒周期でインクリメントしていき、そのカウンタ値を8個のLEDで表示します。プログラムをリスト5-3(稿末)に示します。

LEDデータのセットは、まずCOUNTERレジスタの反転値をLED\_DATAレジスタに入れて74HC164にLEDデータをセットするルーチン呼び出します。LEDセット・ルーチンではLED\_DATAレジスタを左シフトしてそのときのキャリをチェックし、その値を $RA_1$ に出力し、その後CLKを立ち上げてから立ち下げます。この動作を8回繰り返すことで、LED\_DATAの値が74HC164出力の $Q_A$ (LSB)から $Q_H$ (MSB)に出ます。



(a) 回路図



(b) 74HC164のタイム・チャート

< 動作説明 >

- 1回目のCLKの立ち上がりでA入力の状態D<sub>7</sub>をQ<sub>A</sub>に出す
- 2回目のCLKの立ち上がりでA入力の状態D<sub>6</sub>をQ<sub>A</sub>に出すと同時にQ<sub>A</sub>の状態D<sub>7</sub>をQ<sub>B</sub>に出す
- これを8回繰り返すとQ<sub>A</sub>~Q<sub>H</sub>にD<sub>0</sub>~D<sub>7</sub>が出力される

図5-5 2個の出力ポートで8個のLEDを点灯

また、LEDはこの出力が“L”レベルで点灯するので、LED\_DATAの論理値が「0」で点灯となります。なおこの操作をしているときにはLEDがちらつきますが、極力高速で行う、LEDの直列抵抗を調整するなどの工夫が目立たないようにします。

## 5-4 トリマで周波数を調整可能な簡易のこぎり波発生器

ポート(RB<sub>0</sub>)の入力/出力切り替えにより、簡易のこぎり波を作る回路を紹介します。この方法はCRの時間定数回路やRB<sub>0</sub>入力のしきい値を利用して実現します。ただし、こののこぎり波の波形は正確なものではなく、波形の上昇時は曲線となります。

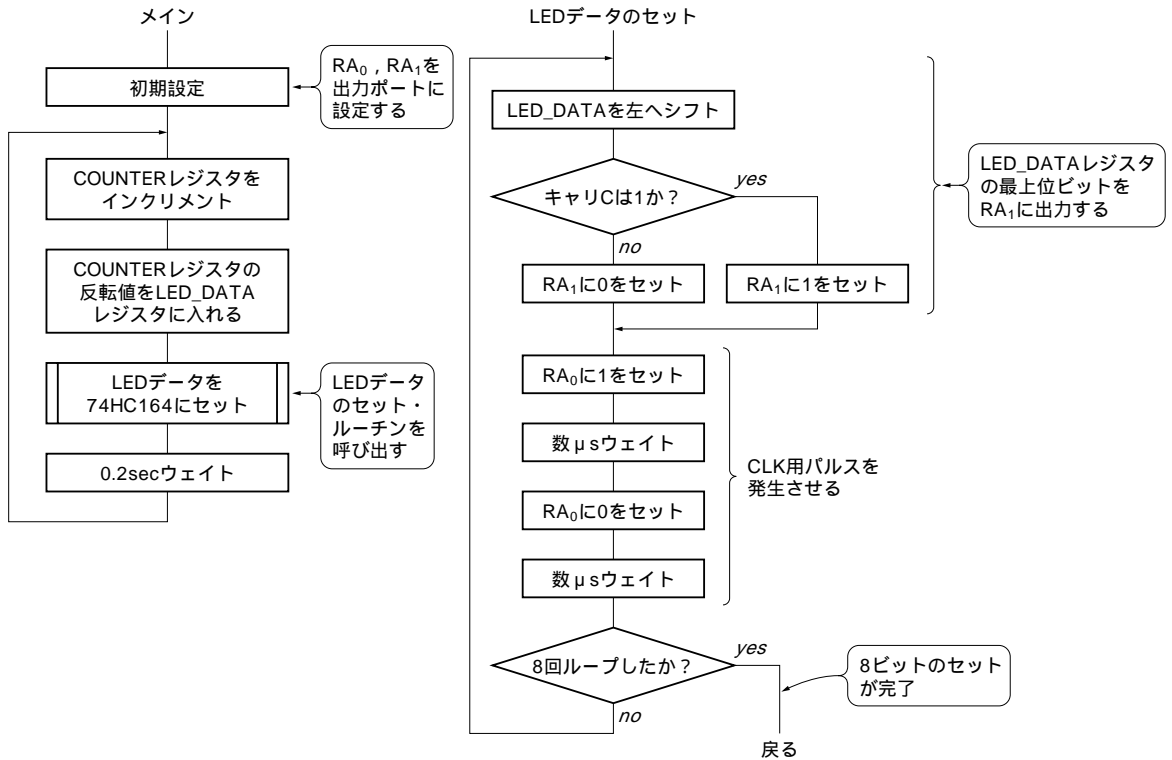


図5-6 8ビット・カウンタ表示のフローチャート

## ● 動作説明

図5-7(a)は回路図です。図のCとRがCR時定数回路です。またダイオードD<sub>1</sub>(1SS133)はコンデンサに蓄電された電荷を瞬時に放電する役割をします。

図5-7(b)の出力波形の各ポイントでの動作を説明します。

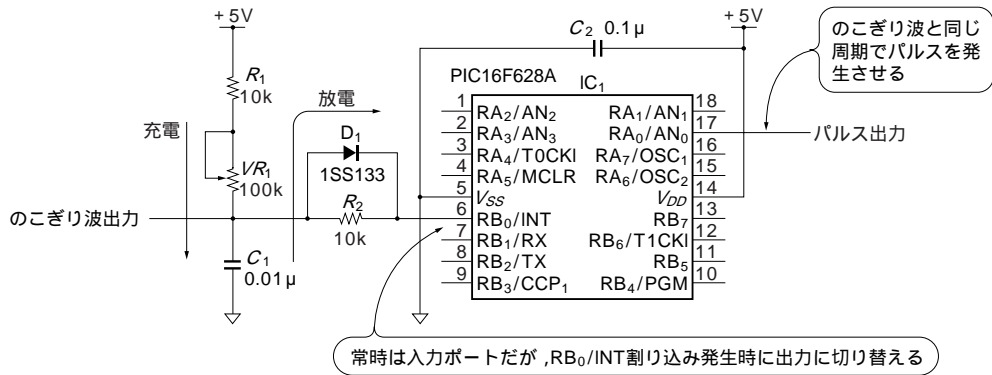
電源ONの直後はRB<sub>0</sub>を出力に設定し、“L”レベルを出力しておきます。

波形の出力開始時にRB<sub>0</sub>を入力に切り替え、RB<sub>0</sub>/INT割り込みを立ち上がり動作で許可します。その後OUTPUTのレベルはCRの時定数で上昇していきます。

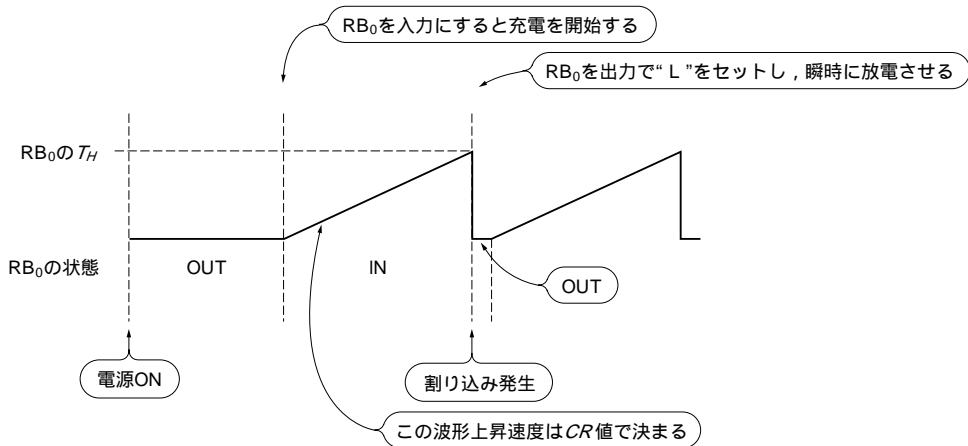
RB<sub>0</sub>ピンのレベルがしきい値に達するとRB<sub>0</sub>/INT割り込みが発生します。割り込み処理ではまずRB<sub>0</sub>を出力に切り替えて“L”レベルを出力します。このときコンデンサCに蓄電された電荷はダイオードD<sub>1</sub>を通して瞬時に放電されます。その後RB<sub>0</sub>を入力に切り替えて割り込みを抜けます。

この動作の繰り返しによりOUTPUTにはのこぎり波に近い波形が出力されます。この波形の周期はCRの時定数で決まるので、Rのトリマ(VR<sub>1</sub>)により周期を調整することが可能です。またOUTPUTは大きな入力抵抗の回路で受ける必要があります。なぜならこの入力抵抗がCRの時定数に影響してしまうからです。

なお、割り込み処理の中でほかのポート、たとえば、RA<sub>0</sub>をセット/クリアすることで同じ周波数のパルスを出力することが可能です。



(a) 回路図



(b) 出力波形

図5-7 簡易のこぎり波発生器

## ● プログラム

図5-8はテスト・プログラムのフローチャートです。まず、初期設定においてRB<sub>0</sub>の立ち上がりエッジでの割り込みを設定し、割り込みを許可します。次に、メインの処理では無限ループで割り込みを待ちます。

割り込み処理ではRB<sub>0</sub>を“L”レベルで出力に切り替えコンデンサを放電すると共にRA<sub>0</sub>にパルスを発生させます。その後RB<sub>0</sub>を入力に切り替えて割り込みを抜けます。このプログラムを実行すると電源オンでのこぎり波が連続出力し、トリマを回すとその周波数が変化します。また、RA<sub>0</sub>からは同じ周波数のパルスが出力します。

プログラムをリスト5-4(稿末)に示します。

## ● 実測値

この回路で実測した結果は以下のようでした。

周波数可変範囲：940 ~ 8900Hz

振幅範囲：0.4 ~ 3.0V