

1-1

PICで“電子工作”から“プログラミング”の世界に!

●簡単な電子工作!! —— PICは安価なワンチップ・マイコン

ワンチップ・マイコンにはいろいろなものがありますが、アマチュアが電子工作用としてもっとも手軽に使えるのがPIC（ピック）と呼ばれるものです。

昨今ではマイコンといえば表面実装形ものが主流で、ディスクリート部品で育ったアマチュアには手に負えません。その点、PICの基本はゲジゲシでおなじみのDIP（ディップ：デュアル・インライン・パッケージ）ですから、アマチュア・レベルのはんだ付け作業で十分に使えます。

PIC（Peripheral Interface Controller）は、米国のマイクロチップテクノロジー社（Microchip Technology Inc.）*が開発したマイクロプロセッサです。一般的には“PICマイコン”と呼ばれて親しまれています。同社のホームページを見るとPICマイコンには図1-1に示したように多くの種類のもので用意されていますが、本書では8ビット・ファミリの“PIC12”と“PIC16”の中からいくつかを選んで使うことにします。

●パソコンは道を走れない! まずは一般的な話から…

家庭で使われているパソコンは、プログラムを取り替えることにより文書作成や表計算、画像処理、インターネットなどいろんなことをやってくれます。

このようにパソコンはなんでもできそうですが、でも自動車のように道を走れるわけではありません。同様に、パソコンにはパソコンに向けた仕事があります。

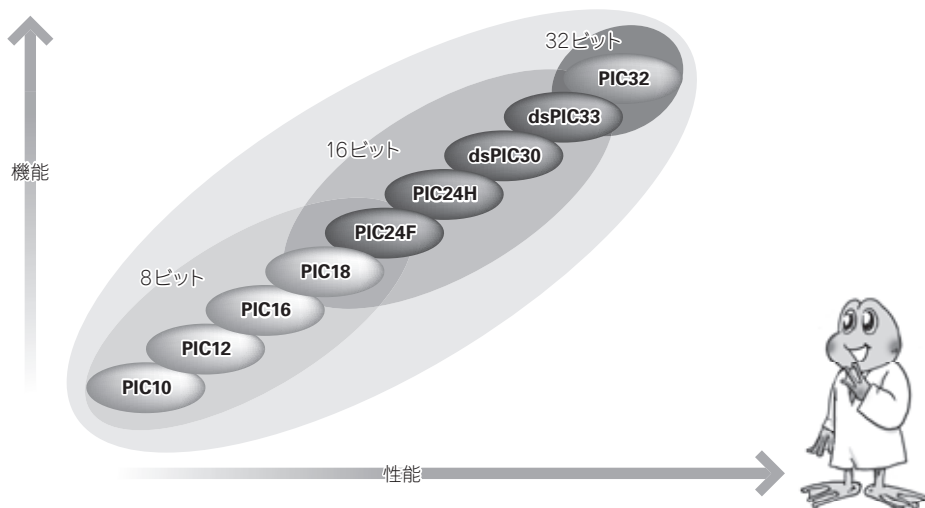


図1-1 PICマイコンのラインナップ（マイクロチップ社のWebページから）

*: <http://www.microchip.com/>

もちろんPICマイコンも、小さくても立派なコンピュータではあるのですが、同じようなことがいえます。電子工作の中にも、PICマイコンに向いているものとそうでないものがあるというわけです。

そのようなわけで、PICマイコンは一般的には電子工作の中でもどちらかというところコンピュータ処理に向けた用途に使われています。

一方では、電子工作の中には実際にはコンピュータでしか実現できないものもあり、PICマイコンがロボット制御などそのような用途に使われてきたのは当然のことといえます。

話はいかかりますが、今まで電子回路を使って電子工作をするとき、これはコンピュータにやらせたほうがうまくいきそうだな、あるいは楽だな、と思ったことが度々ありました。

そこで、本書ではPICマイコンを使ってこの思いに挑戦してみることにします。これは、「PICで電子工作の世界に新風を！」ということです。

●電子回路で作っていたものをPICマイコンで作ったらどうなる？

電子工作の中でもLEDを点滅させるとかタイマのようなものはロジックICやタイマICを使って電子回路で作りますが、これらはどちらかというところコンピュータで処理するのに向いています。でも、ラジオやオーディオ・アンプのようなものはコンピュータでは作れません。

では、今までのロジックICやタイマICで作っていた電子工作を“電子回路方式”，PICマイコンで作るものを“プログラム方式”と名づけて話を進めましょう。

図1-2は、2個のLEDを“電子回路方式”と“プログラム方式”で点滅させる方法を示したものです。この二つは、ともに同じ動作をさせることができます。

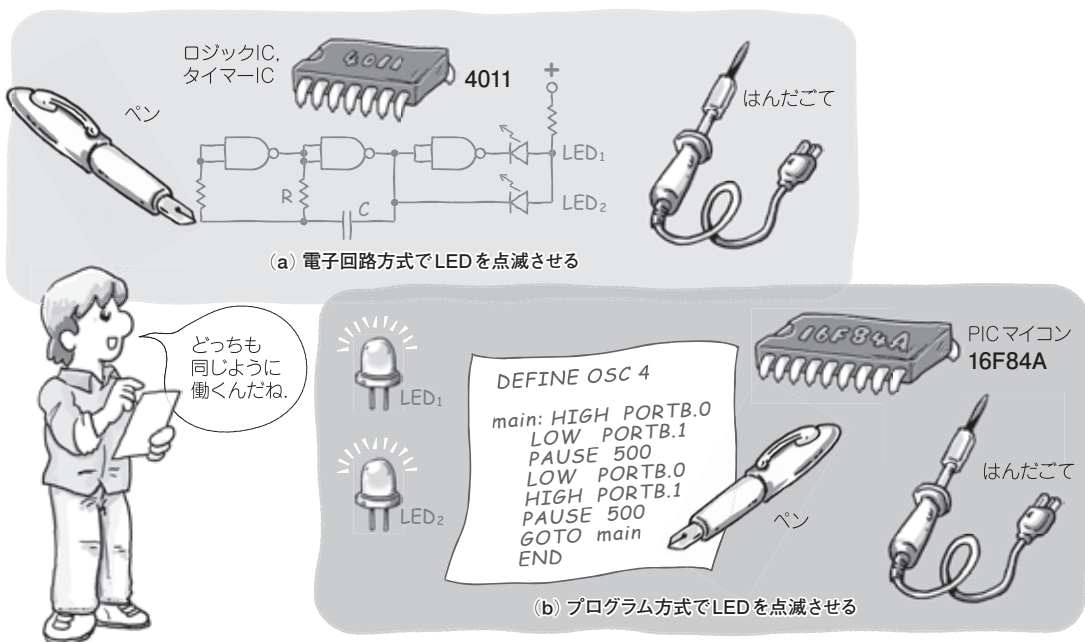


図1-2 電子回路方式とプログラム方式の相違点

2-1-1

電子回路方式 — 基本論理回路を作る

●2入力の論理回路（AND/NAND/OR/NOR）を試してみよう

図2-1は、基本的な論理回路として、2入力のAND、NAND、OR、NORの論理記号と真理値表を示したものです。真理値表をみると分かるように、ANDにインバータをつけたものがNAND、ORにインバータをつけたのがNORです。これら入力の値だけで出力が一意に決まる論理回路のことを“組み合わせ回路”と呼んでいます。

では、図2-1に示した四つの論理回路について、ロジックICを使った電子回路方式と、PICマイコンを使ったプログラム方式で、その動作を体験してみることにしましょう。

●ロジックICで作る論理回路

電子回路方式で論理回路を実現する場合、簡単なものとダイオードを使って作ることもできますが、一般的にはロジックICを使います。

ロジックICにはTTLの74シリーズやCMOSの4000シリーズなどがありますが、ここではもっとも一般的なCMOSの4000シリーズからロジックICを選んでみることにします。

図2-2は、電子回路方式を試してみるために選んだロジックICを示したものです。実は、この4個のロジックICは写真2-1に示したように外観はどれも同じ14ピンDIPで、しかもピン接続がまったく同じなのです。ということは、同じハードウェアで、ロジックICを取り替えることによって4種類の論理回路の実験ができます。

図2-3は実験のための回路を示したもので、(a)は実験回路、そして(b)はハードウェアを組み立てるときの参考図です。

では、(a)の実験回路でその動作を説明してみましよう。この回路では、論理回路の部分を取り

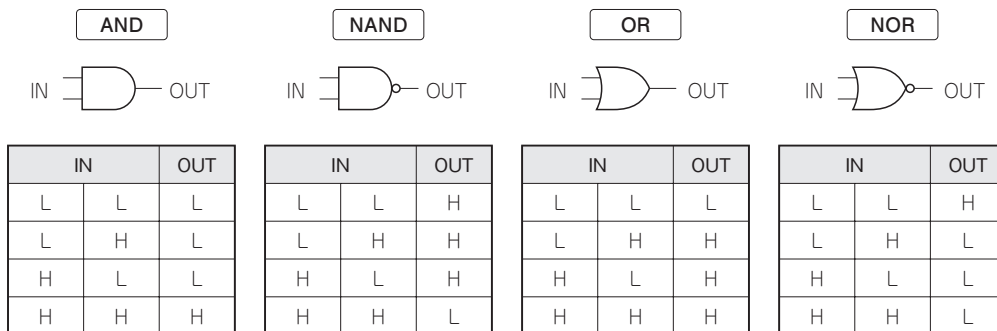


図2-1 2入力の論理回路の図記号と真理値表

替えることによって、図2-1の四種類の論理回路の実験ができます。

まず、論理回路の入力はプッシュONの二つの押しボタン・スイッチ(SW)で行うようになっており、ボタンが押されていないときに“L”，ボタンが押されたときに“H”になります。また、出力はLEDで表示するようになっており、LEDが消えているときに“L”，そして光っているときに“H”です。

実際にハードウェアを組み立てるときには、(a)の論理回路の部分はICソケットを使ってロジックICを差し替えられるようにします。すると、(a)の実験回路では情報が不足で、使用しない入力ピンの処理などをどうしたらいいか分からないところも出てきます。

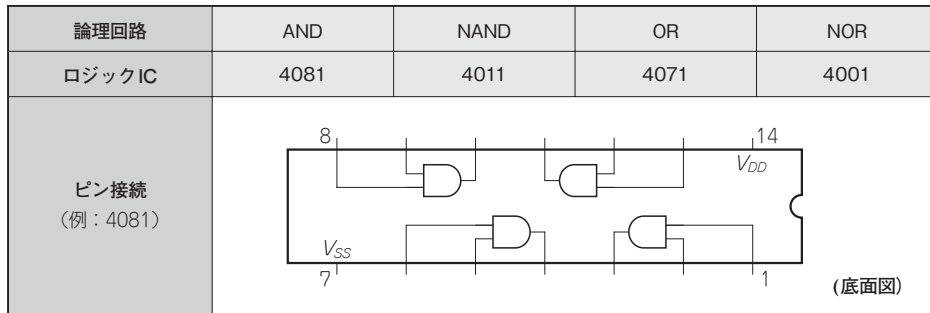


図2-2 論理回路ごとに選んだロジックICとピン接続

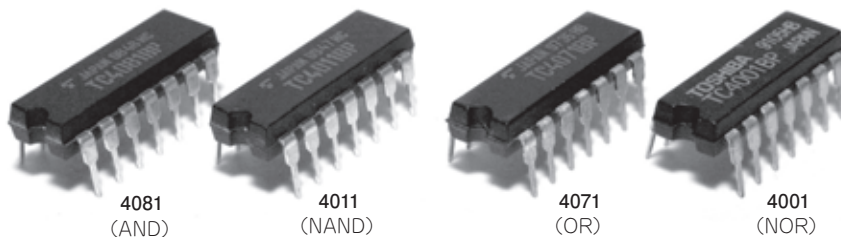


写真2-1 図2-2で選んだ4個のロジックIC

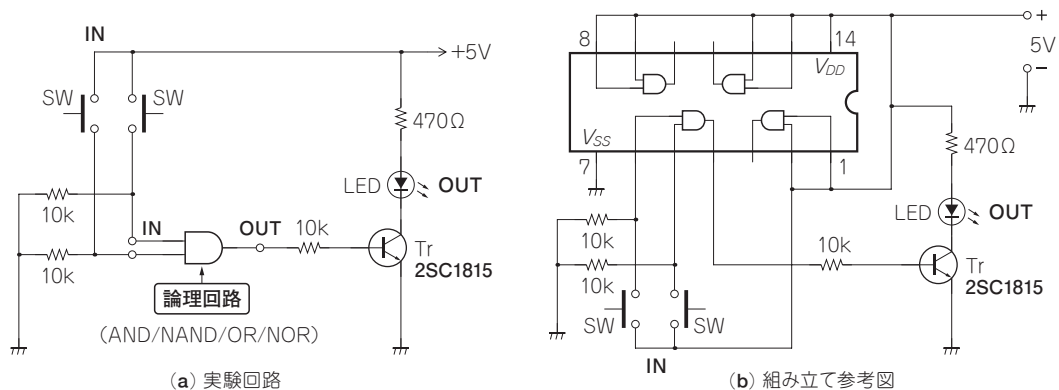


図2-3 電子回路方式で論理回路を実験する回路

●非安定マルチバイブレータでLEDを点滅させる

第2章のT-FFでは、その動作を2個のLEDで確認しました。電子回路方式では、このようにT-FFでも2個のLEDを点滅させることができます。

でも、2個のLEDを点滅する、といった場合にふつうにやるのは、発振器の非安定マルチバイブレータ（ONとOFFを繰り返す回路）による方法です。そこで、ここではロジックICのNANDを使った非安定マルチバイブレータでLEDを点滅させてみることにします。

図3-1(a)が、非安定マルチバイブレータで作る、2個のLEDを点滅させる回路です。LEDを点滅させるだけならスイッチ（SW）は不要ですが、この回路では押しボタン・スイッチのボタンを押したときにLEDが点滅するように作ってあります。

●点滅の周期を変えるには

図3-1では、LEDの点滅速度、すなわち非安定マルチバイブレータの発振周波数は R と C で決まります。その場合、ただLEDを点滅させるだけなら半固定抵抗器（ VR ）は不要ですが、 VR で点滅の速度を変えられるようにしてあります。電子回路方式で作る場合でも、これくらいのことはできます。

では、この回路をサンハヤトのユニバーサル基板（サンハヤト、ICB-90）の上で作ってみましょう。図3-1(b)は、その場合の組み立て参考図です。この参考図を使うと、ユニバーサル基板の上にそのまま作ることができます。

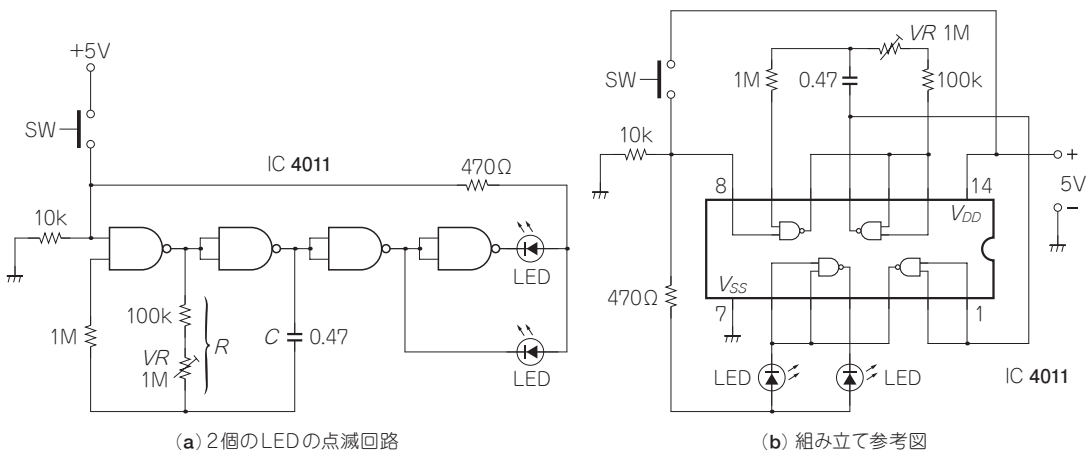


図3-1 電子回路方式で作る2個のLED点滅回路

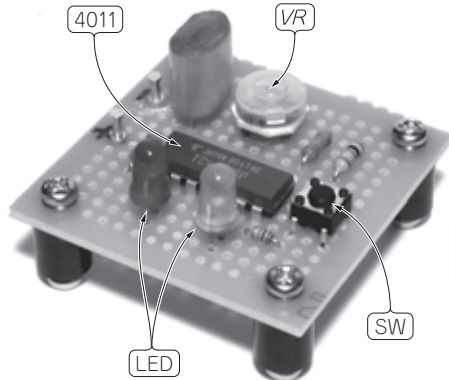


写真3-1 電子回路方式で作った2個のLED点滅装置

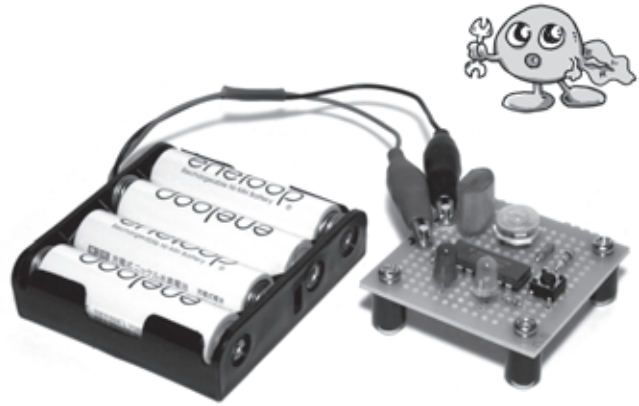


写真3-2 電子回路方式で2個のLEDを点滅させているところ

写真3-1が、電子回路方式で作った2個のLEDの点滅装置です。組み立てが終わったらVRを12時の位置におき、電源を加えてスイッチのボタンを押してみます。すると、2個のLEDが点滅を始めたでしょう。

うまくいったら、LEDを点滅させながらVRを回してみます。これで、LEDの点滅速度が変われば成功です。写真3-2は電子回路方式で2個のLEDを点滅させているところで、スイッチのボタンを押すと、2個のLEDが点滅します。

3-1-2 プログラム方式 — 2個のLEDを点滅させる

●出力を2個もつPICマイコンを選ぶ

PICを使ってプログラム方式でやってみましょう。まずPICマイコンを選んでハードウェアを組み立て、そのあとでプログラムを組みます。

2個のLEDを点滅させる場合、ただ単に2個のLEDを操作するのであれば出力端子が二つあればいいので、PICマイコンには12F629/675を使うことにします。

なお、実際には電子回路方式でやってみた、スイッチのボタンを押している間だけLEDを点滅させる機能と、半固定抵抗器(VR)で点滅速度を変えられる機能を、プログラム方式でもやってみたいと思います。これらの機能を加えても、12F629/675で作れます。

●ハードウェアの組み立て

図3-2が、プログラム方式で2個のLEDを点滅させるためのハードウェアの回路です。

まず、GP₀ピンにつながっている半固定抵抗器(VR)とコンデンサ(C)は、LEDの点滅速度をVRで可変してみるためのものです。LEDの点滅速度は基本的にはプログラムで変えますが、PBPのPOT命令を使うとVRで変えることもできます。

2個のLEDは、GP₁ピンとGP₂ピンにつないで点滅させます。なお、電子回路方式の場合のよう

●音を出す機能のあるPICを選ぶ——16F84A/628A/648A

プログラム方式でトーン(音)をたのしもうとすると、使えるPICマイコンに制限がでできます。その一つは、ちょっと長い曲を演奏させようとするとうプログラムサイズが大きくなるせいか、PICマイコンによっては書き込めない場合があります。また、このあと使うHPWM命令ではPICマイコンにPWM機能が必要なのですが、このような機能は12F629/675や16F84Aには搭載されていません。

というわけで、ここで使うPICマイコンは、18ピンの16F84A/628A/648Aに限定することになりました。

●PWMで音を作る——ハードウェアで作るかソフトウェアで作るか

さて、プログラム方式で音を出したり曲を演奏する場合にはFREQOUT命令やHPWM命令を使います。これらは共にPWM(パルス幅変調)で周波数を作り出すようになっています。そこで、ここにもちょっと配慮が必要になります。

まず、FREQOUT命令はソフトウェアPWMと呼ばれるもので、PBPの命令で周波数を作り出しています。このソフトウェアPWMはPICマイコンの入出力ピンを第2章や第3章で使ってきたI/Oモードで働かせるようになっているので、16F84A/628A/648Aのどれでも使えます。

一方、HPWM命令はPICマイコンの入出力ピンをPWMモードにして使いますが、使い方などその詳細を知るにはどうしてもデータシートを見る必要があります。

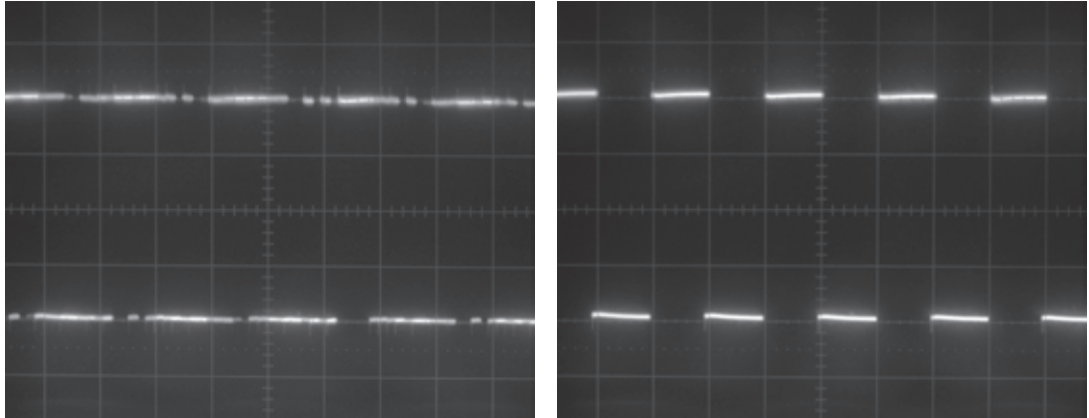
そこで結果だけを記しますと、HPWM命令を使うにはPICマイコンがPWMモードをもっている必要があります。具体的には、PWMモードをもっているのは16F628Aと16F648Aだけなので、PWMモードをもっていない16F84AではHPWM命令は使えません。

16F628A/648Aのデータシートを見ると、図3-7に示したピン接続図のピン9は、表4-1の名前(Name)に示したようにRB₃/CCP₁となっています。このうちのRB₃というのが第2章や第3章で使ってきたもので、CCP₁というのがHPWM命令を使うときのPWMモードの場合です。

表4-1 16F628A/648AのRB₃/CCP₁ピン(ピン番号は9)

名前 (Name)	機能 (Function)	入力レベル (Input Type)	出力レベル (Output Type)	記述 (Description)
RB ₃ /CCP ₁	RB ₃	TTL	CMOS	省略
	CCP ₁	ST	COMS	Capture/Compare/PWM I/O





(a)FREQOUT 命令の場合

(b)HPWM 命令の場合

写真4-1 FREQOUT 命令とHPWM 命令で1000Hzを出したときの出力波形の比較

このように、16F628A/648AのPWMモードではHPWM命令の出力はCCP₁ピン（ピン9）に出できます。なお、PICマイコンによっては複数のPWM出力チャンネルを持ったものもありますが、16F628A/648Aの場合には1チャンネルだけなので、このCCP₁ピンだけに注目していればOKです。

それともう一つ、第4章ではPICマイコンで曲を演奏しますが、曲によってはプログラム・サイズが大きくなるために16F84Aではコンパイルできないという事態になりました。表1-1に紹介したプログラム・メモリの壁を越えてしまったということです。

●FREQOUT 命令とHPWM 命令、どちらを使うか？

このあと、FREQOUT 命令やHPWM 命令を使って周波数を出力しますが、それぞれに特長があって使い方が違います。どのように違うかは実際にそれぞれの命令を使うときに説明しますが、本質的なところをちょっと説明しておきましょう。それは、音質です。

写真4-1は、それぞれの命令で1000Hzの出力を出したときの出力波形を比べてみたものです。これを見ると、(a)のFREQOUT 命令の場合には方形波に細かい不規則な切れ込みが入っており、波形が常に動いていて安定しません。それでもスピーカを鳴らしてみると1000Hzの音はするのですが、どうもがさついた感じの音です。

FREQOUT 命令の場合の出力がこのようになるのは、じつはソフトウェアPWMで周波数を作り出しているせいで、PBPの日本語説明書にはこの命令を使う場合にはローパス・フィルタ(LPF)を入れるように指示されています。また、PBPの英語説明書を見ると、クロック周波数を20MHz以上とすることを推奨しています。そこでFREQOUT 命令を使うときだけクロック周波数を20MHzとします。

その点、HPWM 命令を使った場合には写真4-1(b)に示したようにきれいな方形波になっており、方形波なりの澄んだきれいな音が得られます。

以上の結果だけみればHPWM 命令を使ったほうがよさそうですが、使い勝手は両方で違って、例えばFREQOUT 命令では二つの音を同時に出せませんがHPWM 命令ではそのようなことはできません。このあたりのところは、実際にプログラムを組んで確かめてみることにします。

● タイマIC 555を使った短時間タイマ

まず最初は、「3分間待つのだぞ」でおなじみの、通称「ラーメン・タイマ」を電子回路方式で作ってみます。設定時間はラーメン用の3分以外でも使えるよう、最大5分とします。

図5-1は、タイマICの555で作る短時間タイマの回路です。この回路では設定時間は C と R で決まり、 $C=470\mu\text{F}$ とした場合、 $500\text{k}\Omega$ の VR （可変抵抗器）ではほぼ最大5分になります。

この短時間タイマの回路は実験的なものなので、電源を加えると同時に働き出し、時間がくると電子ブザー（BZ）が鳴ります。そして、この電子ブザーは電源を切るまで鳴り続けます。また、電源を加えると同時にLEDが光りますが、これはあくまでも電源が入っていることを示すパイロット・ランプです。

写真5-1は、図5-1の回路を秋月電子通商のユニバーサル基板Cタイプ（サンハヤトICB-88相当、大きさは $72\times 47\text{mm}$ ）の上に組み立てて働かせてみているところです。

● タイマ時間の校正

この種のタイマでいちばんやっかいなのは、設定時間の目盛りを振る VR の校正です。実は、設定時間を決める C と R のうち、 C のほうは電解コンデンサを使うこととなりますが、入手できるのは静電容量の許容差が $M(\pm 20\%)$ のものでから $470\mu\text{F}$ といっても $376\sim 564\mu\text{F}$ の間にばらついていて、そこで、 VR の校正がかならず必要になります。

というわけで、写真5-1ではまず VR に目盛りを入れるためのラベルを用意しておき、出来上が

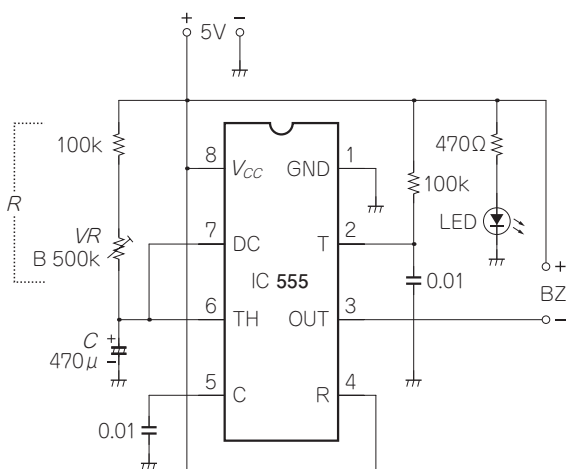


図5-1 電子回路方式で作るタイマIC 555を使ったラーメン・タイマの回路

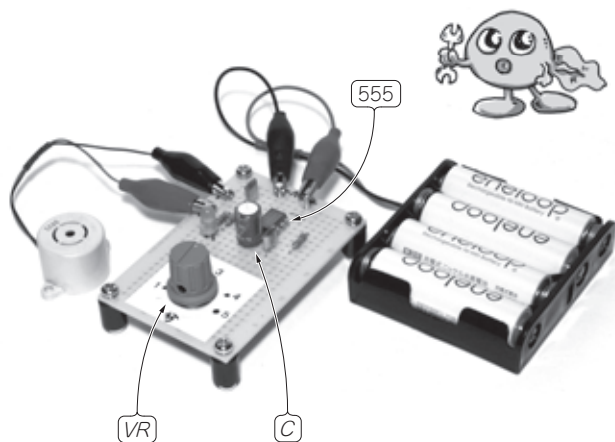
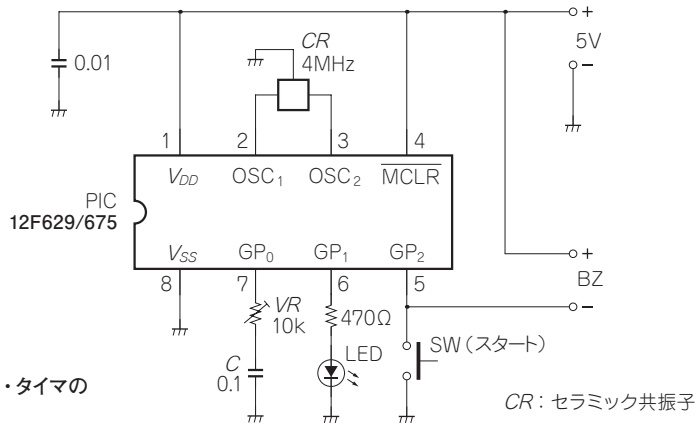


写真5-1 電子回路方式で作ったラーメン・タイマを働かせているところ



図5-2
プログラム方式で作るラーメン・タイマの
ハードウェア回路



CR：セラミック共振子

ったところでストップウォッチとにらめっこで校正しながら目盛りを入れていきます。写真5-1に示した1分ごとの目盛りは、こうやって入れたものです。

この校正は、例えば5分の日盛りを用意するにはストップウォッチで5分間の時間を測らなければならない、しかもそれを何回もやらなくてはならないので時間のかかる作業になります。

5-1-2 プログラム方式——「短時間タイマ」を作る

●8ピンPICマイコンを使ってアナログ的手法でラーメン・タイマを作る

写真5-1に示した電子回路方式の短時間タイマは働きを確かめるだけで、スタートやストップの操作のできない実用性に欠けるものでした。その点、プログラム方式だとプログラムによって実用性を高めることができます。

では、電子回路方式の図5-1に相当する短時間タイマを、PICマイコンで作ってみることにしましょう。短時間タイマは簡単なものですから、使用するPICマイコンは8ピンの12F629/675でやってみることにします。

図5-2が、プログラム方式で短時間タイマを作る場合のハードウェアの回路です。この回路は用意するプログラムとの関連が深いので、プログラムを書くときのことを考えながら回路を説明してみましょう。

プログラム方式のハードウェアは、電子回路方式と違ってスタート用の押しボタン・スイッチ(SW)を用意しました。このSWはGP₂ピンにつながっており、SWが押されていないときには電子ブザー(BZ)によって“H”(1)にプルアップされています。

ここでSWが押されるとGP₂ピンが“L”(0)に落ちますが、プログラムはこれを検出してタイマをスタートさせます。と同時に、スタートの合図として電子ブザーがピッと鳴ります。

●POT命令を使ってVRの値を読み取り、時間の設定をする

電子回路方式の短時間タイマではVRで時間を設定しましたが、プログラム方式の場合にも第3章の図3-3に示したPOT命令を使うと同じようなことができます。