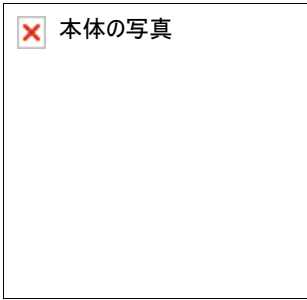


OIDUSラボ:ハード開発課-PICNICの第二回

【公開日】:04/6/25 【著者】:OIDUS・DAVID・DEKURON 【記事コード】:0002 【Ver】:初版

【第二回】:音を鳴らす



皆さん、こんにちは。「PICNIC」第二回です。今回は、PICを使って音を鳴らします。

音を鳴らすと言うことは、スピーカーなどに電流を流すことで実現できるので、何もPICを使うことは無いでしょうが、プログラミングであらゆる波形を作り出せると言う点を考えると、超万能の波形ICが出来上がったようなものなのです。

しかし、デジタルのPICは、やはり2種類の電圧しか出力できません。そうすると、人間に聞こえるいい音を作り出すには、疎密波(0110101101010・・・みたいに)として出力し、アナログローパスフィルタに通すなどの面倒なことをしなくてはなりません。(これは、スーパーオーディオCDの原理です。やることは似てますけどね。)

アナログの波形をダイレクトに出力するにはどうするか？

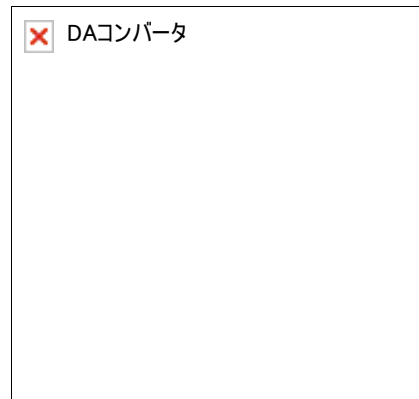
そこで、D/Aコンバータを作って、それを利用することにしました。

D/Aコンバータとは、入力されたデジタル信号の各ビットの重みを、そのまま電源電圧に反映させて、アナログの電圧を出力するものです。

つまり、電源電圧が5Vだとすると、0Vから5Vまでをビット数(分解能)で割り、それをデジタルの値にかけ出力電圧とし、抵抗を使ってデジタルの各ビットごとの出力を足して、最終的なアナログの電圧を得るのです。(足し算回路といいます。)

これを使うと、PICのPORTBあたりから出力される値が、D/Aコンバータを通じて、アナログ出力としてとりだすことができるので、ダイレクトにアナログ波形をスピーカーに流せば、望みどおりの音が出せるのです。

ということで、早速作ってみました。



写真の中の、黒い小さいのが、抵抗です。わかりやすいように、同じ抵抗を直列でつないでみました。入力は4BITです。左のビットに行くほど、抵抗の量が減るので、同じ電圧が通過しても、通過後の電圧は、各ビットで異なります。それを足すと、アナログ電圧となるのですが・・・問題があります。

抵抗は、許容誤差と言うものを含んでおり、抵抗値が製品一個一個で違います。どれくらい違うかは、製品の種類によりますが、今回使った抵抗の場合は最大で±10%も違うのです。これがどれだけ大変なことかと言うと・・・

抵抗値の最小単位は写真赤枠にあるように680Ωです。4BITなので、16段階の分解能です。そうすると、16段階の区別をするには100%÷16で6.25%の違いを区別できなくてはなりません。

もし、3BIT目の合成抵抗値が本来2720Ωであるはずなのに、全て+10%の誤差が出て3808Ωとなってしまったらどうなるでしょうか。電圧は40%狂い、4BIT目の値の精度が無意味なものになってしまいます。

実は、この方法は、ビット数が増えるほど、精度が極端に落ちていくので、あまり高精度な用途には使われません。まあ、例に出した誤差は、あまりに極端ですから、普通は4BITくらいなら誤差など無視しても大丈夫ですね。

別の方法なら、ビット数の増加に対応できますが、今回は圧電スピーカと言う簡単なスピーカを使ったテストですので、ここら辺は無視と言う方向で行きたいと思います！

あと、これ以上ビットを増やすと、今回のような合成抵抗で組むような場合、2のべき乗の割合で抵抗が増えていくので、10BITのコンバータを作るときは10BIT目に至っては1024個もの抵抗をつながなくてはなりません。ひ～。これはすごく、おばかさんなので、抵抗値の違う8種類の抵抗を使って組むのを勧めます。まあ、誤差が平均的になるので、精度は上がるかもしれないですね。かな？

さて、いよいよプログラミングです。

今回は、アセンブラで組むのが面倒なので、Cで組みました。体験版のCコンパイラ「SourceBoost」です。このソフトは統合開発環境になっていて、シミュレーションもできる優れものです。しかし、シミュレータが無かったので、とりあえず、PICを出しているマイクロチップ社の無料アセンブラ「MPLAB」をリンクして、ビルドまでを自動化することにしました。

Cの方は、まずmain関数から始まります。これはお約束ですね。今回、while(1)で永久ループにして、その中でputOTOという、音を出力する(PORTBからデジタル出力する)関数を呼ぶことで一周期のSIN波形を作り出すつもりでしたが・・・やはり、PIC16シリーズではハードの乗算機が備わっていないタイプがほとんどなので、SINのCライブラリが全くありませんでした。

SINはテイラー展開を利用して、級数の形で近似できるので、それを組んでしまおうと思いましたが、今度は少数型のサポートが無くて、できませんでした。

SINの近似解を配列にしておこうかと考えましたが、これもPICのプログラムメモリの少なさと、少数型が無いので、SIN以外の用途(SIN×COSなどの波形を作るなど)に使えないので、需要がありません。おまけに、2バイトまでの変数しかサポートされていない上に、クラス定義もできず、もう八方塞の状態で。

仕方なく、インクリメントしてON・OFFしてTANの「ような」波形を作ることにしました。

原理は、値をインクリメントして、4BITの最大値「15」を超えたら0に戻す、の繰り返し(左図)に、2msのウェイトと、0を挟む。を一周期にして、右図のような波形を作り出しました。



音のほうは、「グイ〜」見たいな音です。あまり心地よいものではありませんが・・・まあ、良しとしましょう。あとで関数をいじればいくらでも波形を変えられますしね。

今後の改良点は、圧電スピーカからクリスタルイヤホンなどのイヤホンに出力先を変えて、ボリュームで音量変更、スイッチで波形の種類を選択、D/Aコンバータの精度向上・・・などでしょうか。

将来的には、この内部波形をレジスタの値に変更して、録音したデータを再生できるようにしたいですね。そうすれば、ICレコーダを作ったり、MP3プレーヤみたいなものも作れます。

そういった入力データの同期を取る方法とデータ読み込みの方法を共通化すれば、いろいろなハードに変身すると思いますね。

でもそれには、まず他のPICを書き込めるライターをつくらないと・・・

リンク

[OIDUSタウントップ](#)

[PICNICトップ](#)