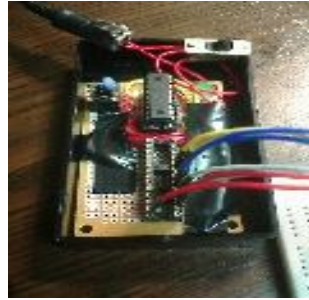


OIDUSラボ:ハード開発課-PICNICの第一回

【公開日】:04/6/24 【著者】:OIDUS・DAVID・DEKURON 【記事コード】:0001 【Ver】:第二版

【第一回】:標準回路ボードを作る



第二版。更新しました。更新部分は青色で表示します。

注意を呼びかける内容は赤色で表示します。

皆さん、こんにちは。「PICNIC」第一回です。PICNICは、「PICマイコン」なるものを使って、電子工作を行っていくページです。

尚、今回は、第一回ということもあり、PICについての簡単な解説も設けました。「PICマイコン」を知っている方には「本題」部分からお読みになることをお勧めします。知らない方に簡単に解説。

PICマイコンとは、その名の通り、「マイコン」の一種です。マイコンとは、早い話、「超小型コンピュータ」だと思っちゃってください。「見かけ」は「IC」です。あの足が何本も生えた、黒いのです。丁度、上にある写真の真ん中にある長方形の黒いのがそれです。

で、その黒いマイコン、電源にさすと、中にあるコンピュータが、これまた中に入っている「プログラム」を実行します。これは、WINDOWS用なり、MAC用なり、UNIX用なり、漢字TALK用なり・・・とにかくパソコンに入ってるソフトをクリックして実行するのと本質的に違いはありません。

そして、それは「CPU」の仕事です。CPUは、プログラムにかいてある内容どおりに、仕事を進めていきます。たとえば、「足し算する」だったり、「掛け算する」だったり。

欠かせないものに「メモリ」がありますね～。メモリが無いと、動画編集をしたり、写真編集したり、3Dゲームをしたり・・・やりたいことを満足にできません。

さて、こう言った、パソコンを触った人なら誰もが一度は聞いたことのある大切な部品・・・コンピュータを動かすのに必要なものを、コンパクトなICサイズに収めたのが「マイコン」なのです。中にはCPUも入ってます。メモリも入ってます。後は電源だけ！（ザウルスC1の宣伝みたいですね・・・）

具体的に、このマイコンから生えている足、「ピン」と言いますが、その中で、電源にさすものが2本（プラス・マイナス）あるんです。それを電源につなぐとマイコンが動き出します。でも、くれぐれもコンセントにささないように！乾電池2本から4本分くらいの電圧で動くので、4本分の電圧を越えると、故障します！（これは、製品によって異なります。各々の詳細な情報は、「データシート」と言うものから得られますので、参照してみるのも良いでしょう。）

責任は取れませんので、最低限「コンセントにつなぐ」のはやめてくださいね。

電源につなぐだけで、プログラムが動き出すのです。こんなに簡単なものはありません。ただ、プログラムが動くだけでは「何の役にも立ち」ません。マイコンの中で何が起ころうと、知ったことではないのです。パソコンだったら、モニタに結果を表示したり、スピーカーから音を出したりしますが、もし、これが全く無かったら、中で何が起ころうと、それを確認することはできませんよね。(見れないですし…。CPUファンのいまいましい雑音くらいは聞こえますかね。)

そうなのです！知るために、「ピン」をつけているのです。ピンからは、プログラムのメモリの内容を「電流」として流したり、マウスのような「入力」を、これまた電流として流し込みます。その結果を、プログラムが利用したりするのです。ピンからは電流のやり取りしかなされません。これは、誰がなんと言おうと、そうであるのです。だって、金属の導線の中を流れるのは電気くらいですから。

この「電流」を「うまく」、「制御」して液晶に流すと、液晶に文字が表示されたり、スピーカーに電流を流せば当然音が出ます。それをプログラムで行います。

そして、マイコンのすごいところは、「プログラム」が実行できること自身にあるのです。

現在、プログラムと言うものの、大半が「パソコン＝パーソナルコンピュータ」用に作られると言ってよいでしょう。しかし、逆に、パソコンが無ければ動きません。正確に言うなら、「CPU」と、プログラムが利用する「機械」をそろえないと満足には動かせません。たとえばマウスだったり、サウンドカードだったり。CPUで動くプログラムを「ソフトウェア」といったりします。

一方、その対極にあるのは「ハードウェア」です。ハードウェアは、CPUなどの特定の動作しかしないような部品には依存しません。ハードウェアは、専ら「機械」をさします。機械とは、特定の動作をする、「物理的な」ものです。

そして、ハードウェアは「動く」機械です。(モータなど、動く部品がくっついてると言う意味ではありません。制御ができるという意味です)

ソフトウェアがCPUの中で動く「論理的」なものであるのに対して、ハードウェアはその反対に位置します。

しかし、機械を作るのは、すごく大変です。指先一本で無限大の世界を開ける可能性をもったソフトウェア作りと違い、ハードウェア作りは電子部品の「温度特性」「抵抗」その他もろもろの、物理的な、部品個々の「性質」を考慮に入れなければなりません。そして、もちろんそれは難しい…。特に、特定の動作をする機械(FMラジオとか)を作るには、「電子回路」というものを組まなければならない、ソフトウェア作りに比べて、非常に手間のかかるものでした。

まずそういった状況を打開するために作られたのが「IC」です。ICは、特定の動作をする部品です。それは、今までばらばらになっていた電子部品を組み合わせる作業(回路作り)を、使いやすい「道具」として、「ある程度」まとめたものです。これにより、ものすごく機械作りが楽になり、効率もあがりました。

さて、そんな中、ふっと「デジタル」が登場します。いままで「アナログ」でやっていた回路作りを、「ロジック」と言うもので置き換えたり、新たに作り出したりしていきました。

ロジックとは、「ブール代数学」と言う数学の一分野に基づき、機械上でいろいろな挙動を制御する手法の総称です。そして、それを実現するには、3種類の記号で、いろいろな計算を実現する…難しい分野なんですよ…。

とにかく、ブール代数とは、3種類の計算「NOT・OR・AND」と言うもので、数値を操作する学問

です。なんと、あらゆる計算は、この3種類の記号と数値の組み合わせで表現できるらしいです。

これを利用すると、あらゆる数学の計算を、たった3種類の記号と数値の組み合わせで表現できます。(くどい!)そして、電子部品でこの3種類の記号を回路にすることで、回路上で数学を動かせるようになったのです。つまり、数学を機械で操作できるというわけ。この「記号」を実現する回路には、超がつくほど有名な「半導体」と言うものを主に使っています。これまた有名ですが、「ダイオード」や、「トランジスタ」と言った半導体の性質を利用して、この3つの記号を回路にします。たとえば、「AND」の記号だったら、ON・OFFのスイッチをつなげて作ります。

しかし、ICやら電子部品やらで回路を組んでいくには限度があります。部品にはある程度の大きさが必要ですので(もちろん、はんだ付けのレベルでの話ですが)、複雑な回路になると、必要な大きさに収めきれなかったり、ICが増えすぎるなどの理由で、線がごちゃごちゃになったり、ICに欲しい回路がのっていないなどの問題が発生してくるわけです。

これらの問題は、「FPGA」や「ASIC」、「CPLD」などの手法である程度、解決できるようになりました。コンピュータ上で回路をかいたり、プログラミング「のようなもの」で回路を表現したものを、IC(やLSI以上の素子)に転写(書き込み)するような形で、オリジナルの「ICチップ」を作れるようになったのです。必要な回路が「ワンチップ」で実現できる…効率や、精度の向上につながりました。

よって、アナログな回路では実現しがたかった、小型な、電卓などの計算機や、「計算(演算)」をする機械…つまり「コンピュータ」を作り出すことができるようになったのです。それらは、一気に世の中に広まっていきました。

コンピュータの唯一無比の特性は、「CPU」があることでしょう。CPU上で動くプログラムは、完全に「論理」の中で動きます。これは、ハードウェア作りとの決定的な違いでしょう。

そんな中、このマイコンが登場しました。マイコンは、先ほど述べたように「コンピュータ」です。つまり、プログラムが動かせます。そして、見かけICです。いやむしろ、完全にICです。

これは何を意味するのか…

つまり、「ハードウェア」と「ソフトウェア」の架橋的な存在なのです。物理的な問題を解決する手間のかからない「ソフトウェア作り」と、動く機械作りができる「ハードウェア作り」をいっぺんにできるのです。そして、どちらかと言うと「ハードウェア作り」に利用されることが多いです。

ICなどを使った機械を作る上で、ロジックや電子部品の組み合わせで回路を実現する場合と、プログラミングでハードを作るマイコンとでは、手間やコストなどの観点では、明らかにマイコンにぶがあるでしょう。

そして、そういったマイコンの「一種」が、「PIC」なのです。PICは、いろんな本やらネットのページやらで情報があふれていますから、初心者でもとっつきやすいです。また、プログラミングをやっていた人には、最初のうちは、あまりハードウェア作りと言うことを意識せずにはじめることができるでしょう。(いずれハードを学ぶときが来るでしょうが…)

「アセンブラ」を学ぶのにも、もってこいです。CPUの動作やメモリとの入出力など、プログラミングの技術力向上にもつながります。やって損は無い、それがPICなのです!

【本題】


さて、長々と説明してきましたが、いよいよ本題です。PICを使う上で、最低限必要になる回路は、電源周りの処理と、IOポート(IOピン)と電子回路との接続、クロックの入力、リセット、そのく

らいでしょうか。

これらの「絶対必要になる」機能を、毎回毎回、一から作ってるのは無駄な努力(ただし、もちろんデバッグの段階での話ですが。)ですので、それらを使いやすい「ボード」にしまおう、というのが、今回のテーマです。

★ちなみに、今回の回路図はフリーソフトの「回路図エディタBSch Version 0.90」を使用させていただきました。いや～、本当便利です。ドラッグドロップで部品を配置して、BMPファイル出力までしてくれる優れものです。おまけに、「VC++」などでパーツのライブラリを作れるらしいので、今度作ってみようかな～★

早速作りましょう。まず、ボードとする以上、わざと回路上の経路を切っておく場所があります。入出力は「PIC16F84A」ではPORTAで5ピン、PORTBで8ピンですので、以下の図の説明のように、ピン全てを丸ピンであけておきました。ここに、部品を差し込むだけで、回路になります。そして、VDDIにつないで出力とするか、VSSにつないで入力とするかを、切り替えスイッチで切り替えられます。また、クロックの入力も丸ピンであけておきました。リセットは、プッシュスイッチでリセットとします。

 回路図

さて、実際に使ってみると、かなり便利です。もともと、一号機を作っていたので、今回は2号機になります。

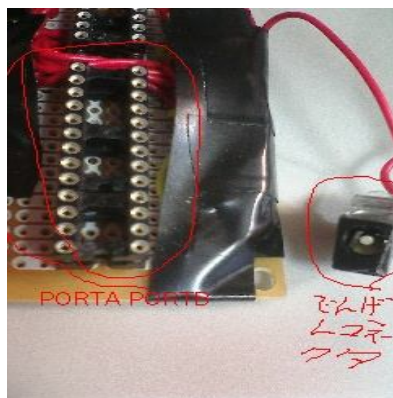
一号機は、配線処理が極めて雑で、空中配線の嵐と言った感じでした。もう芸術的です。(写真ではあまり伝わりにくいでしょうが・・・)



何より、電源コネクタは無く、直接乾電池をつないで、リード線の先にはんだをつけて硬くしたものを挿して使っていました。PORTAは無く、入力もできず、しょっちゅう回路の電源が落ちて、もう放電を起こしているような状態でした。



2号機の写真も載せておきます。



このボードは、今でも現役です。かなり効率が上がりました。特に、ブレッドボードも安物で、ジャンパ線も8本しかないので、助かります。それに、ブレッドボードに直接つなぐよりも動作が安定しますね。多分。(GNDを一本化したのが功をそうした様で、無駄にGNDとつながなくても、ひとつでもつなげばOKです。)

LEDなんかを挿す場合、丸ピンと言うことで直接させます。これは便利です。

クロックなんかは、動作周波数はいつも20MHzで固定なので、挿しっぱなしでよく、ほとんどPICの差し替えでOKです。あとは、抜き差し無く書き込めるようになれば便利ですがね～(いっそ、そうしましょうか！切り替えスイッチで書き込みピンを制御するとか、書き込みには3ピンくらいしか使わないですからね。)

ただ、ケースがプラケースとはいえ、加工が面倒で、今は裸で使用しています。早くカッコいいケースに入れたいです。

今、各種のピン数のPICに書き込みのできるライタの製作記事を見つけたので、それに挑戦中です。一応、ライタ機能に、今回の標準ボードの拡張として、40ピンまでのPICに対応したボードをドッキングした便利な作品にする予定です。記事の作者の方の許可をいただければ、PICNICの連載に載せますので、まあ、待っててくださいな。

おしまい。

リンク

[OIDUSタウントップ](#)

[PICNICトップ](#)