

ハードウェアとソフトウェアのトレードオフを考えた設計とは

# デジタル・クロック設計の 過程と部品の選定

見  
本

今回製作するデジタル・クロックを例に、設計の過程について順を追って説明します。設計のやり方にもいろいろあると思いますが、本書では入手可能な部品を選ぶところから始めます。

ハードウェアの設計とはいえマイコンを使う場合は、この段階でソフトウェアのこともある程度考慮しておく必要があります。

その後、実際の製作の手順についても簡単に説明します。

## はじめに

ハードウェアは、できるだけ使用する部品を少なくし、製作を楽に、また部品費を安く上げて作ることが基本です。しかし、今回のように私たちが少量の物を作る場合は、家電などの量産品と違ってそれほどシビアに考える必要はないでしょう。また、ソフトウェアでは面倒だと思われる場合は、多少部品が増えなくてもハードウェアで実現させるのも一つの手です。

それから、部品の入手性も重要なポイントです。場合によっては、入手した部品に合わせて設計することも必要になります。

## 6-1 何をさせるかを考える

### 仕様、機能の検討

まず、製作する装置で何をさせるかを考えます。最初に必要な機能を洗い出し、それらをハードウェアとソフトウェアのどちらで分担させるかを考えます。ソフトウェアを使う場合は、それをどのように実現させるかもある程度は考えておく必要があります。

この本ではデジタル・クロックを製作しますから、まず、時を刻むのに正確な1秒周期のクロック(1Hz)が必要になります。時刻を表示するには、7セグメントLEDかLCD(液晶表示器)のようなものがが必要です。さらに、時刻を合わせるためには、操作用のスイッチも必要です。時計ですから、電池で駆動することも考えます。

- ▶ 1秒周期の正確なクロック
- ▶ 時刻を表示するもの

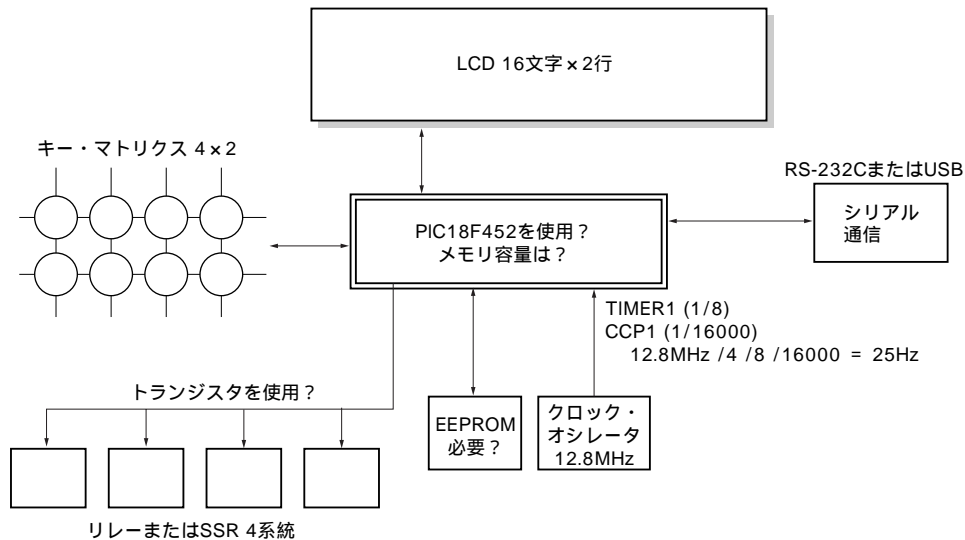


図6-1 ラフ・スケッチ

全体の構成を考える。実際は手書きでメモ程度のもを何度か書いて、少しずつ詳細を決めていく。設計に入る前に、I/Oポートの数や使用するペリフェラルの内訳をある程度決めておく。

- ▶ 時計合わせ用のスイッチ
- ▶ 電池で駆動するためには、効率の良い安定化電源

さらに、タイマ機能を付けることにしているので、外部に接続した機器を制御する部分も必要です。何を制御するかにもよりますが、一般的なものでは充電器や照明、ラジオなどのAC100V機器を制御することが多いでしょう。

- ▶ 100Vをスイッチングできる機構(リレー<sup>①</sup>やSSR<sup>②</sup>)
- ▶ 制御は2~4系統

このアイコンは、章末に用語解説があります

その他、必要な機能を挙げてみます。

- ▶ 電源が切れてもタイマ時刻(予約プログラム)を保持できるようにEEPROMを使用
- ▶ 拡張機能としてパソコンからタイマ予約する場合の非同期シリアル通信ポートまたはUSBポート

仕様は、初めから厳密に決めることは意味がありません。なぜならアマチュアの場合、仕様に合わせて部品を選べるわけでもなく、むしろ、部品に合わせた仕様になる、というのが実状だからです。後から変わることもある、という前提でラフに決めておきます(図6-1)。

## マイコン・チップの選定

仕様に基づいてI/Oポートの数や必要なモジュール、外部に必要な回路などを検討します。この時点で、使用するPICデバイスを仮に特定しておきます。

最初はI/Oポートの数やPICに内蔵されているモジュール(ペリフェラル)の機能に注目します。パソコンとシリアルで通信する場合はUSARTがついているほうがよいし、EEPROMも内蔵のもので容量が足りるか、足りなければ外部に接続するためにI<sup>2</sup>Cなどのインターフェースも必要になる、ということを明確にします。また、どのI/Oポートで何の機能を使うかといったことも具体的に決めていきます。設計を進

表6-1 デジタル・クロック I/O マップ 18F452 用

ピンと機能の割り当て表。どのピンに何の機能を割り当てるかを明確にするために設計時に記入する。また、プログラムのコーディング時、機能変更/拡張などの再設計時にはI/Oマップとしても使用する。

|   | ピン番号 | I/O    | 用途   |
|---|------|--------|--|
| RA <sub>0</sub> /AN <sub>0</sub>                    | 2    | IN/OUT | 未使用  |
| RA <sub>1</sub> /AN <sub>1</sub>                    | 3    | IN/OUT | 未使用  |
| RA <sub>2</sub> /AN <sub>2</sub> /V <sub>REF-</sub> | 4    | IN/OUT | 未使用  |
| RA <sub>3</sub> /AN <sub>3</sub> /V <sub>REF+</sub> | 5    | IN/OUT | 未使用  |
| RA <sub>4</sub> /T0CKI                              | 6    | IN/OUT | 未使用  |
| RA <sub>5</sub> /AN <sub>4</sub> /SS/LVDIN          | 7    | IN/OUT | 未使用  |
| OSC <sub>1</sub> /CLK                               | 13   | IN     | クロック・モジュール入力                               |
| OSC <sub>2</sub> /CLKO(RA <sub>6</sub> )            | 14   | IN/OUT | 未使用  |
| (RB <sub>0</sub> )INT <sub>0</sub>                  | 33   | IN     | 未使用  |
| (RB <sub>1</sub> )INT <sub>1</sub>                  | 34   | IN     | 未使用  |
| (RB <sub>2</sub> )INT <sub>2</sub>                  | 35   | IN     | 未使用  |
| (RB <sub>3</sub> )CCP <sub>2</sub> *                | 36   | IN     | 未使用  |
| (RB <sub>4</sub> )                                  | 37   | IN     | キー・パッド Col <sub>4</sub>                    |
| (RB <sub>5</sub> )PGM                               | 38   | IN     | キー・パッド Col <sub>3</sub> (Pic-Key PGM)      |
| (RB <sub>6</sub> )PGC                               | 39   | IN     | キー・パッド Col <sub>2</sub> (Pic-Key PGC)      |
| (RB <sub>7</sub> )PGD                               | 40   | IN     | キー・パッド Col <sub>1</sub> (Pic-Key PGD)      |
| (RC <sub>0</sub> )T1OSO/T1CKI                       | 15   | OUT    | リレー・ポート P <sub>1</sub> (LED <sub>1</sub> ) |
| (RC <sub>1</sub> )T1OSI/CCP <sub>2</sub> *          | 16   | OUT    | リレー・ポート P <sub>2</sub> (LED <sub>2</sub> ) |
| RC <sub>2</sub> /CCP <sub>1</sub>                   | 17   | OUT    | 未使用  |
| RC <sub>3</sub> /SCK(SCL)                           | 18   | OUT    | I <sup>2</sup> C SCL                       |
| RC <sub>4</sub> /SDI(SDA)                           | 23   | IN/OUT | I <sup>2</sup> C SDA                       |
| (RC <sub>5</sub> )SDO                               | 24   | IN/OUT | (Pic-Key ICD)                              |
| RC <sub>6</sub> (TX)CK                              | 25   | OUT    | シリアル TX                                    |
| RC <sub>7</sub> (RX)DT                              | 26   | IN     | シリアル RX                                    |
| (RD <sub>0</sub> )PSP <sub>0</sub>                  | 19   | OUT    | リレー・ポート P <sub>3</sub>                     |
| (RD <sub>1</sub> )PSP <sub>1</sub>                  | 20   | OUT    | LCD E                                      |
| (RD <sub>2</sub> )PSP <sub>2</sub>                  | 21   | OUT    | LCD R/W                                    |
| (RD <sub>3</sub> )PSP <sub>3</sub>                  | 22   | OUT    | LCD RS                                     |
| (RD <sub>4</sub> )PSP <sub>4</sub>                  | 27   | IN/OUT | LCD Data <sub>0</sub>                      |
| (RD <sub>5</sub> )PSP <sub>5</sub>                  | 28   | IN/OUT | LCD Data <sub>1</sub>                      |
| (RD <sub>6</sub> )PSP <sub>6</sub>                  | 29   | IN/OUT | LCD Data <sub>2</sub>                      |
| (RD <sub>7</sub> )PSP <sub>7</sub>                  | 30   | IN/OUT | LCD Data <sub>3</sub>                      |
| (RE <sub>0</sub> )RD/AN <sub>5</sub>                | 8    | OUT    | キー・パッド Row <sub>1</sub>                    |
| (RE <sub>1</sub> )WR/AN <sub>6</sub>                | 9    | OUT    | キー・パッド Row <sub>2</sub>                    |
| (RE <sub>2</sub> )CS/AN <sub>7</sub>                | 10   | OUT    | リレー・ポート P <sub>4</sub>                     |

めるうちに、I/Oや内蔵モジュールの過不足などでPIC選定に見直しが必要な場合もあります。

表6-1は設計時にI/Oを定義するとき使用するピン・アサイン・マップです。ピン名の欄は複数の機能のどの信号を使うかで信号名を丸で囲むなどして印をつけます。I/Oの欄は、そのピンが入力か出力かを明記します。用途は文字通り、そのピンを何の用途に使用するかを記述します。これは具体的に「リレー1をONにする」とか「ブザーを鳴らす」というように記述したほうがよいでしょう。一通り表を完成させた後で、I/Oの過不足によりほかのチップが使えないかを再検討します。

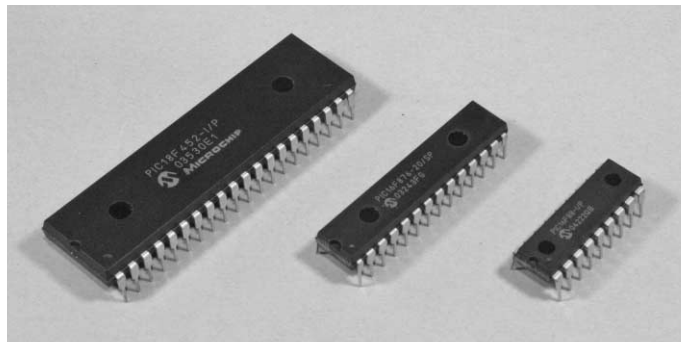


写真6-1 PICパッケージの外観  
プラスチック・パッケージでフラッシュ・メモリ・タイプのPIC。左から18F452(40ピン)、16F876(28ピン)、16F88(18ピン)

この表を設計時にまとめておくことで、I/Oの重複が防げ、過不足が一目瞭然となります。また、プログラムを作成するときにもI/Oマップとして使い、ミスも減らせます。後から機能を追加する場合や、これを元に新たに設計する場合などは、こういった資料はなおのこと重要です。

マイコン・チップを選択する際は、ICパッケージの大きさ(40ピンや28ピンなど)や入手性の良さ、価格なども考慮します。また、開発環境が安価にそろうということも重要です。でも、実際は好みや使い慣れたもので選ぶ場合もあります(写真6-1)。

## プログラム容量

マイコンを使う場合、I/Oや内蔵モジュールのほかにプログラム容量についても考えておかななくてはなりません。こういう処理があるから何Kバイト必要、というのは作り方にもよるので、なかなか難しいところですが、何回か作っているうちに、この処理は何KバイトのPICで収まりそうだな、ということは何となくわかるようになります。

この本では読者の方が独自にまったく違うものを製作するという事も想定していますので、時計を作るにはかなり役不足ですが、18F452を選定しています。プログラム・メモリの容量が大きいので、C言語でプログラムを作成して、Pic-Key(ICD: In Circuit Debugger)を使ってデバッグする場合も余裕があると思います(表6-2)。

もし、プログラムを作っていて容量が足りなくなったら、ひたすらむだな処理を削っていくしかありません。こういう状況になると、プログラムは非常に見にくくなります。

表6-2 メモリ容量の比較

これはデバイスごとのメモリ容量の比較表。プログラム・メモリのサイズは、ワード数で比較している。

|                    | 16F873 | 16F876 | 16F877 | 18F452 | 18F442 | 18F252 | 18F242 |
|--------------------|--------|--------|--------|--------|--------|--------|--------|
| プログラム・メモリ<br>(ワード) | 4K     | 8K     | 8K     | 16K    | 8K     | 16K    | 8K     |
| データ・メモリ<br>(バイト)   | 192    | 368    | 368    | 1536   | 768    | 1536   | 768    |
| データEEPROM<br>(バイト) | 128    | 256    | 256    | 256    | 256    | 256    | 256    |
| パッケージ(Pin)         | 28     | 28     | 40     | 40     | 40     | 28     | 28     |

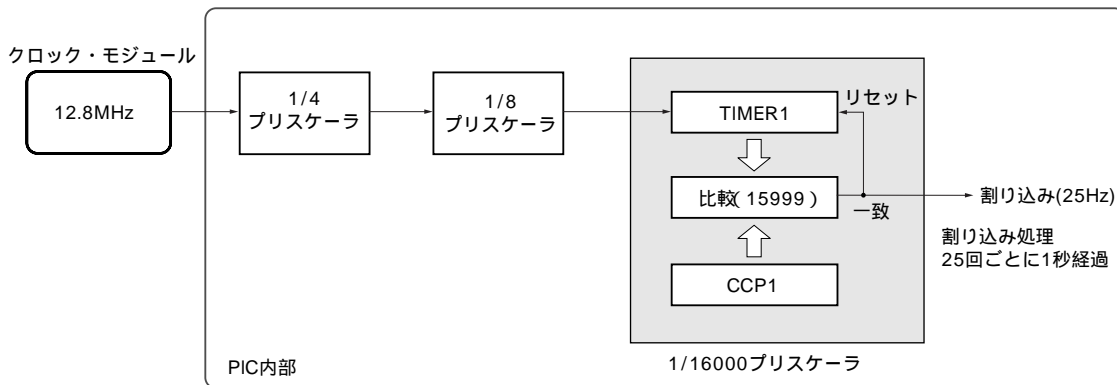


図6-2 1秒の生成  
12.8MHzのクロックから1秒周期を作成する仕組み。CCP1はコンペア・モードで使用する。

### 時計用1秒の生成方法の検討

漠然とクロック・モジュールを使うことは決めています，具体的にどうやって1秒という時間を作るかということは，回路構成やPIC内蔵モジュールの関わりもあるので，最初に決めておかななくてはなりません．時計にとっては，ここが一番重要なところです．

TIMER0をフリー・ランさせて，オーバフロー割り込みを使用する方法もありますが，今回はTIMER1とCCP1(コンペア・モード)を組み合わせで時計用の1秒の元を作ることになります．

PICの動作クロックを12.8MHzとしたとき，TIMER1で内部クロックを使用するように設定すると，周波数は初段で無条件に1/4になります．また，プリスケータは最大の8(1/8分周)に設定します．TIMER1は16ビット・カウンタですが，これをCCP1のコンペア・モードと併用します．コンペア・レジスタCCPR1に15999を設定してTIMER1を16000カウントごとにクリアするように動作させると，CCP1+TIMER1で1/16000のプリスケータとして働きます．このプリスケータを含めて最終的な周波数を計算すると，

$$12.8\text{MHz} \times 1/4 \times 1/8 \times 1/16000 = 25\text{Hz}$$

となります．このCCP1の割り込み(比較一致)を利用すると，1秒間に25回割り込みが発生します．逆にいうと，CCP1割り込みを25回数えることで正確な1秒という時間を得ることができます．ソフトウェアのところでも説明しますが，割り込み処理で25回につき1回カウントされる秒のソフトウェア・カウンタを作り，これを数えることで時間を刻みます．

このようにTIMER1+CCP1を使った場合，TIMER0のオーバフロー割り込みよりも割り込みの頻度が少なくなるため，割り込み処理のオーバーヘッドがかなり軽減できます(図6-2)．

## 6-2 具体的な部品の検討

大まかに何をやらせるかを決めたら，具体的にどういう部品が使えるか，何が必要か，その部品は簡単に入手できるか，費用はどれくらいかかりそうか，といったことを検討していきます．設計のやり方にもいろいろあると思いますが，電子機器メーカーなどの人たちと違って，私たちは簡単に入手できる部品を使うというのが大前提となります．

まずは，部品の情報や資料を集めることから始めます．トランジスタ技術誌の広告や通販ショップの

Webサイトなどで、販売されている部品を調べるのがよいでしょう。最近では半導体や機構部品などのデータ・シートが、各メーカーのサイトで無償で手に入ることが多いので、部品に見当がついている場合は、多少時間がかかっても資料をそろえることが容易になりました。部品は安くてよいものを探すのが腕の見せ所です、ここが大変ですが、楽しい(?)工程でもあります。

表6-3(a) 主な使用部品の部品表(メイン・ボード)

| 部品番号   | 部品名称           | 型番など                         | 説明, 注意事項  | メーカー             | 備考      |
|--|----------------|------------------------------|---|------------------|---------|
| U <sub>7</sub>   | 三端子レギュレータ      | 7805(ノーマル)<br>TA4805S(低ドロップ) | 出力電圧5V, 500mA以上のレギュレータを使用。電池で使う場合はロー・ドロップ・タイプを使うほうがよい   | 東芝など             |         |
| JK <sub>1</sub>  | DCジャック         | MJ-179Pなど                    | 基板取り付け用のDCジャック。センタ・ピン径は2.2mm。使用するACアダプタに合わせる  | マル信無線電気          |         |
| C <sub>3</sub> , C <sub>14</sub>   | 電解コンデンサ        | 16V以上, 100μF以上               | 耐圧が16V以上の一般的なもので可。ACアダプタと電池を切り替えて使う場合は, 470μF以上の大きいものを使用したほうがよい   |                  |         |
| C <sub>8</sub> , C <sub>9</sub>  | 積層セラミック・コンデンサ  | 0.1μF                        |   |                  |         |
| U <sub>1</sub>   | EEPROM         | 93C46                        | FT232BMのシリアル番号などの識別用ROM。同一のUSBパスに複数のFT232BM搭載機器を接続するときが必要   | マイクロチップ・テクノロジーなど | USB     |
| U <sub>2</sub>   | USB-シリアル変換チップ  | FT232BM                      | USBとシリアル信号を変換するチップ。0.8mmピッチのQFP32パッケージ。U <sub>6</sub> とU <sub>2</sub> は同時に実装できない                                  | FTDI             | USB     |
| X <sub>1</sub>   | セラミック発振子       | 6MHz                         | FT232BMのクロック発生用   |                  | USB     |
| CN <sub>4</sub>  | USBコネクタ(B)     |                              | Bタイプの基板取り付け用USBコネクタ   |                  | USB     |
| LED <sub>3</sub>   | LED            |                              | USBで送受信があった場合に点灯する通信モジュール用LED   |                  | USB     |
| R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub> , R <sub>4</sub> ,<br>R <sub>6</sub> , R <sub>7</sub> , R <sub>9</sub> , R <sub>11</sub> ,<br>R <sub>13</sub> , R <sub>14</sub> | 1/4Wカーボン抵抗器    |                              |   |                  | USB     |
| C <sub>1</sub> , C <sub>2</sub> , C <sub>10</sub>  | 積層セラミック・コンデンサ  | 0.1μF                        |   |                  | USB     |
|  | LCD(液晶表示器)     | SC1602B S*L                  | 16文字×2行   |                  |         |
| VR <sub>1</sub>  | 半固定抵抗器         | 10k ~ 20k                    | LCDの濃さ調整用ボリューム  |                  |         |
| CN <sub>5</sub>  | ヘッダ・ピン・ソケット    | 14P                          | LCD用  |                  |         |
| U <sub>3</sub>   | PIC            | PIC18F452                    | 40P DIPパッケージ。14ビット版の16F877とピン配置に互換性があるため、差し替えが可能(ただし、プログラムに互換性はない)。PICライターで書き込むときは、外す必要があるため、ICソケットを使用する          | マイクロチップ・テクノロジー   |         |
| U <sub>5</sub>   | クロック・モジュール     | 18S-83A-4<br>12.8MHz         | 高精度の正弦波発振器。デジタル回路用のため出力はクリップされているので、完全な正弦波ではない。このクロックは時計の1秒を作るために使用し、PICのシステム・クロックとしても使用する                        | 京セラ              |         |
| R <sub>15</sub> , R <sub>16</sub>  | 1/4Wカーボン抵抗器    | 10k                          | 交流出力を直流出力レベル・シフトさせるための抵抗器   |                  |         |
| U <sub>6</sub>   | RS-232Cレベル変換IC | ADM232AAN<br>SP232ACPなど      | TTLレベルの信号をRS-232Cレベルへ変換するIC。互換品が多数あるので、どれを使ってもよい。U <sub>2</sub> とU <sub>6</sub> は同時に実装できない                        |                  | RS-232C |
| C <sub>4</sub> , C <sub>5</sub> , C <sub>6</sub> , C <sub>7</sub> ,<br>C <sub>13</sub>   | 積層セラミック・コンデンサ  | 0.1μF                        |   |                  | RS-232C |
| CN <sub>7</sub>  | ナイロン・コネクタ      | DF1B-3P-DSA                  | RS-232C信号用の基板側コネクタ。必要に応じて使用。勘合するコネクタ(ケーブル側)DF1B-2S-2.5R。コネクタを使わずに直接配線してもかまわない。RS-232Cを使う場合は、はんだ付けタイプのD-SUBコネクタが必要 | ヒロセ              | RS-232C |