このPDFは，CQ出版社発売の「ARMマイコン パーフェクト学習基板」の一部分の見本です．
内容・購入方法などにつきましては以下のホームページをご覧下さい．
<http://shop.cqpub.co.jp/hanbai/books/MTR/MTRZ201009.htm>

8/16ビット・マイコンのように手軽に使える32ビット低消費電力マイコン

ARM Cortex-M3マイコンLPC1343の特徴と付属基板のセットアップ

プロローグ

桑野 雅彦 / 編集部
Masahiko Kuwano

本書は，ARM Cortex-M3 コア・マイコン LPC1343（NXP セミコンダクターズ）の入門書です．

全体を二つに分けて，前半の BASIC 編では BASIC インタープリタを使ったマイコン・プログラミングを，後半の C 言語編では無料で使える開発ソフトウェア LPCXpresso IDE を使った C 言語によるマイコン・プログラミングを解説します．

このプロローグでは，付属基板の特徴，LPC1343 と ARM Cortex-M3 コアの概要，特集内容を実験するための付属基板のセットアップについて説明します．

付属基板の四つの大きな特徴

● ドラッグ&ドロップでプログラムを書き込める

多くのマイコンはプログラムを書き込むために，専用のアダプタ（書き込み器）や書き込みソフトウェアが必要です．

LPC1343 はこれらが不要です．付属基板を USB ケーブルで Windows PC と接続すればプログラムを書き込めます．

プログラムの書き込みは**図 1** のように，Windows 上でプログラム・ファイルを付属基板のマイコンのフラッシュ・メモリにドラッグ＆ドロップするだけです．

● センサ IC/OP アンプ IC などが実装されている

付属基板の外観を**写真 1** に示します．付属基板には，ARM Cortex - M3 マイコン LPC1343 のほかに，SPI バス対応リアルタイム・クロック（カレンダ）IC PCF2123，I^2C バス対応温度センサ IC LM75B，タッチ・スイッチ IC PCF8883T，OP アンプ IC SA5234D，基準電源 IC TL431A を搭載しています．別途拡張基板などを用意する必要はありません．

本書では，ARM マイコン LPC1343 とこれらの IC を使って，温度測定，カレンダ・データの表示とアラーム，タッチ・スイッチ入力，電圧測定などを学習します．付属基板の回路図を**図 2**（p.8）に示します．

● BASIC インタープリタで楽々プログラミング

本書では，手軽にマイコンを学習するために，開発ソフトウェアのインストールなどを必要としない BASIC インタープリタ「Micro BASIC」を用意しました．

Micro BASIC は**図 3** ように，ターミナル・ソフトウェア（通信ソフトウェア）で簡単に利用できるインタープリタ言語です．文法も分かりやすいのですぐにマイコン・プログラミングの世界を体験することができます．

● LPCXpresso 基板の LPC－Link を繋いでデバッグ

付属基板にはデバッガ機能がありません．しかし，C 言語で開発する場合はやはり，デバッガを利用できると便利です．付属基板は，LPC1343 マイコン評価キット LPCXpresso（**写真 2**．別売り．コラム参照）の LPC - Link が使用できます．

見本

面倒くさい環境構築からおさらば

Micro BASICで楽々プログラミング！

第1章

桑野 雅彦
Masahiko Kuwano

なぜ Micro BASIC なの？

図1を見てください．Micro BASIC インタープリタを起動した後，プリセットされている LED 点滅プログラム・リストを表示して動かし，動作を中断させてプログラムが格納されている領域の先頭 16 バイト分の内容を 16 進数で 1 バイトずつ表示させてみた例です．

少々変わった記号による記述などもありますが，IF，GOTO，GOSUB，PRINT など，なんとなく動作がイメージできるのではないかと思います．

```
******** Micro BASIC for PEACH-1343 ********

OK.
LIST
1000 REM P0.7 LED BLINK
1010 Q=$50008000 Q(0)=Q(0) | $80
1020 P=$50000200 D=0 GOSUB 1900
1030 I=0
1040 I=I+1 IF I<1000 GOTO .
1050 GOSUB 1900 GOTO 1030
1900 P:0)=D D=D^$80 RETURN

OK.
RUN

!BREAK
[Line#1040]

OK.
NEW

OK.
1000 A=.& I=0
1010 PRINT$ A:I) " " I=I+1 IF I<16 GOTO 1010
RUN
E8 03 20 41 3D 2E 26 20 49 3D 30 00 F2 03 20 50
OK.
```

図1　Micro BASIC の実行画面．統合開発環境の煩雑さは皆無

● 必要なソフトウェアはターミナル・ソフトウェア

BASIC 編では，付属基板の ARM Cortex - M3 マイコン LPC1343 を，この簡易言語 Micro BASIC を使って動かすことを主体としてみました．

LPC1343 と Windows PC を USB で直結し，PC 側ではハイパーターミナルなどのターミナル・ソフトウェアを起動するだけで簡単にプログラムの作成や I/O などの操作が行えます．一般的なマイコン開発では必須となるような開発環境のインストールや書き込み器の用意，C 言語の習得なども必要ありません．

Micro BASIC は命令の数もごく少ないので，第 3 章のリファレンスを見ればすぐに理解できるでしょう．言語仕様が単純なだけに，大規模なプログラムを作るのには向いていませんが，ちょっとしたポート・アクセス，データの読み出しや表示などを行うために必要な最低限の機能は備えているので，マイコン応用の機器をちょっと動かしてみたいといった用途には十分に使えると思います．

ソース・コードも公開しており，IAR 社の統合開発環境（IDE）の無償版（評価版）でも再ビルド可能なので，独自のコマンドを追加してもよいでしょう．

● C 言語ではなく，あえて Micro BASIC

通常のマイコンの入門であれば，まず統合開発環境と C 言語というのがあたりまえのようになっています．本書で扱う LPC1343 にしても，各社から統合開

準備は5分もかからない！

Micro BASICをARMマイコンにドラッグ&ドロップ

第2章

桑野 雅彦
Masahiko Kuwano

付属基板を動かそう

それでは，付属基板を動かしていくことにしましょう．ボード上のD1部分にLEDを付けておくとあらかじめMicro BASICに組み込んだプログラムを実行したときに点滅します．プロローグを参考にLEDを付けておきましょう．

付属基板を動かすためには，ターミナル・ソフトウェア(通信ソフトウェア)を使用します．Windows XP，またはそれ以前のWindowsであればハイパーターミナルが添付されていましたが，Vista以降は削除されてしまいました．Vista以降で使用する場合は，フリーのターミナル・ソフトウェアとして有名なTera Termが便利です．Tera Termの入手方法，インストール方法は付属CD-ROMを参照してください．

ここでは，Micro BASICインタープリタ起動までの要点を説明します．セットアップの詳細はプロローグや付属CD-ROMを参照してください．

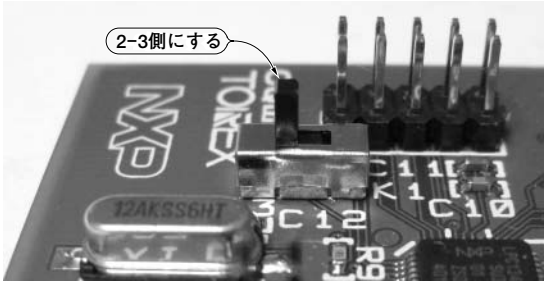


写真1　付属基板のK1を2-3側にする

● USBケーブルで接続

付属基板のK1を2-3側にします(**写真1**)．付属基板をUSBケーブルでWindows PCのUSBポートに接続します．

USBメモリを最初に接続したときと同じようにデバイスが認識されて，USBマス・ストレージ・ドライバが組み込まれます．これは1回目だけで，2回目以降は単に接続すればすぐに認識されます．

● ドライブがオープンする

USBケーブルを接続してWindowsに認識されてからしばらくすると(20秒程度かかることもあるので慌てずに待つ)，エクスプローラが立ち上がり，**図1**のようにfirmware.binというファイルが表示されます．

この中身はフラッシュ・メモリの中身そのものです．firmware.binというファイル名はLPC1343が勝手に付けたもので特に意味はありません．

● ファイルfirmware.binの差し替え

このfirmware.binをいったん削除します．通常のファイル操作と同じように選択して削除します．

続いて，Micro BASICインタプリタの実行ファイルである，LPC_MBASIC.bin(付属CD-ROMの「Micro BASIC binary」にある)をドラッグ&ドロップします(**図2**)．これで書き込みは終了です．

● シリアル・ポート・ドライバの組み込み

いったんUSBケーブルを抜きます．付属基板の

覚えることが少なく手軽で簡単

Micro BASIC の文法と使い方

第3章

桑野 雅彦
Masahiko Kuwano

Micro BASIC とは

Micro BASIC は，VTL(Very Tiny Language)を参考にして作成した簡易言語と，BASIC 風の表記方法の両方をサポートした簡易言語です．

VTL とは，整数型 BASIC を記号化したような言語です．VTL では，例えば BASIC の，

GOTO 行番号(指定した行番号にジャンプする)

を，

#= 行番号

と表現するわけですが，見慣れない人にはとっつきにくい感じがするだろうと考えて，BASIC 風の表記も使えるようにしています．両方を混在させて利用することもできますが，適当にどちらかを削除してしまってもよいでしょう．

使用できる文字

英文字は大文字/小文字とも受け付けますが，コマンドなどでの区別はありません．

例えば，

GOTO

Goto

goto

などは，すべて同じ命令として扱われます．文字列表示のときは，大文字は大文字，小文字は小文字で表示されます．

プログラム領域と1行のサイズ

Micro BASIC のプログラムは行が並んだものです．プログラム領域は全体で 4K バイト，1 行の長さは 80 バイトまでです．

プログラムの入力とダイレクト・モード

Micro BASIC 起動後やプログラムの実行を終了したときはコマンド待ち状態になっており，1 行入力されるのを待ちます．1 行の終わりは[Enter]キーです．入力された文字列の先頭が数値か否かによって動作が分かれます．

- 入力された行の先頭が数値のとき

行番号とみなしてプログラムを格納します．行番号の小さい順に自動的に整列されます．

- 入力された行の先頭が数値以外のとき

コマンドとして即座に処理します．プログラムとして保存せず直接実行されるため，ダイレクト・モードと呼びます．

＜例1＞

```
1000 A=0
900 B=3
950 C=2
```

と入力してから，

LIST(プログラム・リストの表示コマンド)

LEDの点滅とタッチ・スイッチ入力で学ぶ

汎用入出力GPIOの使い方

第4章

桑野 雅彦
Masahiko Kuwano

マイコンとLEDやスイッチといったマイコンの外の世界との接点となるのがI/O(Input/Outpu)ピンです．I/Oピンは「I/O端子」や「入出力端子」などと呼ばれることもありますが，いずれも同じ意味です．

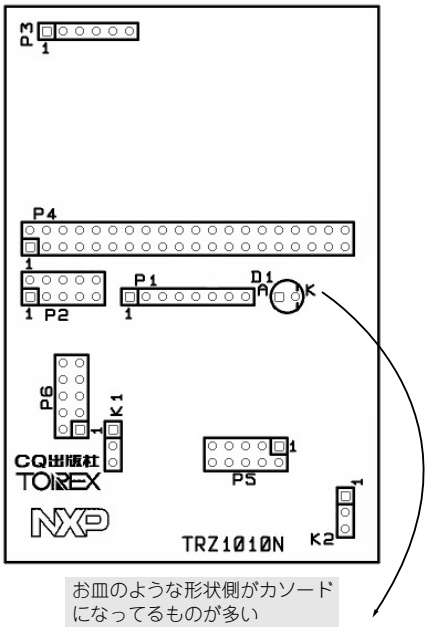
I/Oピンは大きく分けて，

- タイマやシリアル・ポートなどの内部にあるI/O機能のための信号端子
- CPUからのデータ・リード/ライトでデータを入出力する信号端子

の二つの機能に分けられます．

特に後者の機能については，CPUからの制御によって各端子ごとに自由に状態(HighかLowか)を設定できたり，逆にピンの状態を読み出すことができる機能ということで，データシートなどではGPIO(General Purpose Input/Output：汎用入出力端子)と表現されることが多いようです．LPC1343のデータシートでも，やはりGPIOと表現されています．

本章では，Micro BASICでLPC1343のGPIO機能を利用してみます．



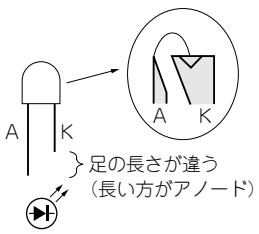


図1　LEDの取り付け位置と方向

GPIO出力機能の実験

GPIOは入力用，出力用のいずれにも利用可能ですが，まずはLEDを使って出力機能を試してみることにしましょう．

● LEDの取り付け

図1のように，D1の位置にLEDを取り付けてください．

LEDには極性があり，一般的に図のように足の長い方がアノード(A)，短い方がカソード(K)電極です．基板を図の向きに置いたときにA(アノード)が左側，

PWM 波形出力で学ぶ
タイマ / カウンタの使い方

第5章

桑野 雅彦
Masahiko Kuwano

LPC1343 には，16 ビット・タイマ/カウンタと 32 ビット・タイマ/カウンタが内蔵されています．

16 ビット・タイマ/カウンタとしては CT16B0 と CT16B1 の二つが内蔵されており，32 ビット・タイマ/カウンタとしては CT32B0 と CT32B1 の二つが内蔵されています．

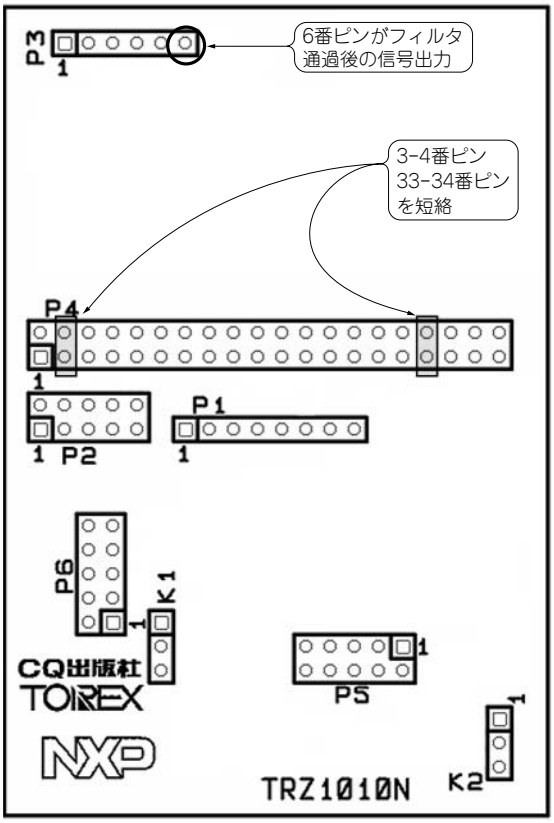


図1　タイマ動作テスト時のジャンパ・ピン設定

CT16B0 は Micro BASIC で使用しているので，ここでは，CT16B1（16 ビット・タイマ/カウンタ 1）を使って 100Hz から 10kHz までの方形波をステップ的に周波数を変えながら出力してみましょう．

付属基板のジャンパ設定

付属基板のジャンパ・ピン設定を**図 1** に示します．**図 1** は必要最小限の設定だけを示していますが，ほかの章で行っているテスト用のジャンパ設定があってもかまいません．**図 1** の回路の主要部を**図 2** に示します．

PWM 波形の出力プログラム

● PWM 波形出力プログラムの動作

リスト 1 は，16 ビット・タイマ/カウンタのテス

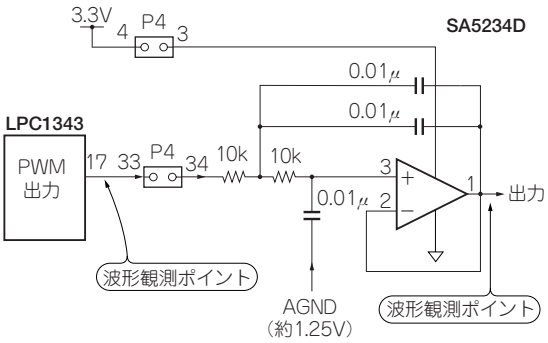


図2　タイマ動作テスト時の回路

基準電圧源でA-D変換値を補正

A-Dコンバータの使い方

第6章

桑野 雅彦
Masahiko Kuwano

LPC1343内蔵A-Dコンバータの特徴

● A-Dコンバータの役割

温度や湿度，明るさなど，自然界の情報は単に‘1’か‘0’かではなく，連続的に変化するアナログ信号です．

マイコンの用途としては，単にONかOFFかといった2値信号の入出力だけではなく，このようなアナログ的に変化する情報を取り込んで処理したいという場合も数多くあります．

A-Dコンバータ(Analog to Digital Converter)はアナログ電圧を数値化するもので，「A-D変換器」や「ADC」，単に「AD」などと表記されることもあります．

例えば，0～3.3Vの間を変換する符号なし8ビットのA-Dコンバータであれば，0Vから3.3Vの間を均等に256(= 2^8)分割した値が得られます．入力電圧が1.65Vであれば128(3.3 ÷ 1.65 × 256)となりますし，逆に読み取られた値が$40(=64)であれば，入力電圧は0.825V(3.3 ÷ 256 × 64)と算出できるわけです．

● LPC1343内蔵A-Dコンバータの特徴

LPC1343にもA-Dコンバータが内蔵されており，付属基板とMicro BASICでも利用可能です．

LPC1343の内蔵A-Dコンバータの基本仕様は以下の通りです．

- 分解能：10ビット
- 変換方式：逐次比較型
- 最小変換時間：2.44 μs
- 入力チャネル：最大8チャネル

このほか，サンプルでは使用していませんが，次の2点もLPC1343のA-Dコンバータの特徴と言えるでしょう．

- 複数の入力チャネルを連続して変換するバースト変換モードを持つ
- 直近の変換データを保持するレジスタとチャネルごとの変換データを保持するレジスタを両方持っている

付属基板のジャンパ設定

付属基板のジャンパは，**図1**のように設定します．このときの回路は**図2**のようになります．

紹介するサンプルでは，LPC1343のA-D変換チャネル3，5，6(AD3，AD5，AD6)を使用します．ジャンパ設定でチャネル3にAGND(アナログ・グラウンド．約1.25V)，チャネル5と6にはAGNDを基準としてOPアンプで約11倍に増幅した電圧を与えます．

I²C，SPI 接続の IC に簡単アクセス

温度センサ IC とリアルタイム・クロックICからデータを読み出す

第7章

桑野 雅彦
Masahiko Kuwano

温度センサから温度データを読み出す

● I²C 対応温度センサ IC LM75B

付属基板にはディジタル温度センサ IC LM75B (NXP セミコンダクターズ)が搭載されています．

付属基板のジャンパを**図 1** のように設定すると，LM75B は I²C バス・インターフェースでマイコンと接続され，LPC1343 内蔵の I²C コントローラを使って LM75B にアクセスできるようになります．

LM75 の使い方や I²C バスのアクセス方法の詳細は第 11 章を参照してください．基本的な操作方法は全く同じです．

● I²C バス・コントローラのレジスタの初期化

Micro BASIC の温度データ取得プログラムを**リスト 1** に示します．

1010 行と 1020 行が初期化部分です．Micro BASIC 起動時には I/O ピン設定などは終わっているので，初期化は I²C バス・コントローラ関係のレジスタのみです．

P(6)でステータス・クリアを行い，P(4)と P(5)に SCL 信号の Low 期間と High 期間の幅(PCLK のクロック数)を与えます．

Micro BASIC 動作時の PCLK は 66MHz に設定しているので，660 を設定すると 10 μs(=1/66 × 660)になります．High 間，Low 期間ともに 10 μs なので周期は 20 μs になり，ここからビット・レートは 50kbps(=1 ÷ 20 μs)となります．

I²C バスとしてはかなり遅い動作ですが，今回はターゲットが温度センサだけで，しかもそれほど大量のデータ伝送をするわけでもないので問題はないでしょう．

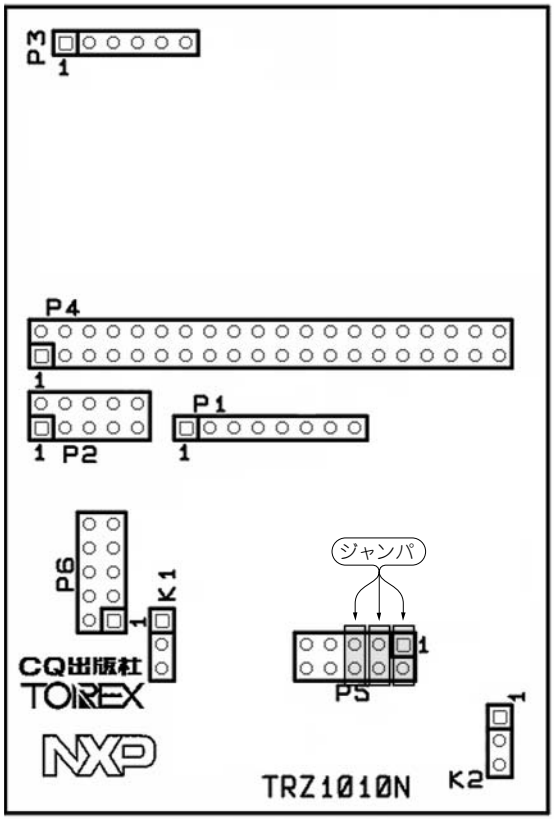


図 1　LM75B テスト用ジャンパ設定

LPCXpresso IDEとC言語による

A-Dコンバータの操作方法

第8章

桑野 雅彦
Masahiko Kuwano

第6章では，Micro BASICを使ったA-Dコンバータの操作を紹介しました．

本章では，C言語からのアクセス・サンプルということで，LPCXPressoの環境からアクセスしてみることにします．

基本的なアクセス方法は，Micro BASICからの場合と変わりありません．レジスタの詳細などは，第6章で説明しているのでそちらを参照してください．

C言語の場合には，構造化できることや構造体が使えること，CMSIS(Cortex Microcontroller Software Interface Standard)で内部レジスタの定義が行われているなどの利点があります．

A-Dコンバータのレジスタの定義とアクセス方法

A-Dコンバータなどのレジスタの定義は，

CMSISv1p30_LPC13xx/inc/LPC13xx.h

に記述されています．各I/Oの構造体を定義してあり，実際のアドレスをこの構造体へのポインタとして定義しています．

例えば，A-Dコンバータであれば，**リスト1**のようにA-Dコンバータのレジスタ構造が構造体で定義されています．

アドレスは**リスト2**のように定義されており，またLPC_ADCというポインタが**リスト3**のように定義されているので，以後は，

dat = LPC_ADC->GDR;

という具合に，構造体へのポインタとしてアクセスすればよいわけです．

A-Dコンバータの操作手順

基本的な操作は次のようになります．

① SYSAHBCLKCTRLレジスタでA-Dコンバータのクロック供給を開始
② PDRUNCFGレジスタでパワーON状態にする
③ PIOをアナログ入力ポートに設定

リスト1　A-Dコンバータのレジスタが定義された構造体

```
typedef struct
{
  __IO uint32_t CR;
  __IO uint32_t GDR;
       uint32_t RESERVED0;
  __IO uint32_t INTEN;
  __I  uint32_t DR0;
  __I  uint32_t DR1;
  __I  uint32_t DR2;
  __I  uint32_t DR3;
  __I  uint32_t DR4;
  __I  uint32_t DR5;
  __I  uint32_t DR6;
  __I  uint32_t DR7;
  __I  uint32_t STAT;
} LPC_ADC_TypeDef;
```

リスト2　A-Dコンバータのアドレスの定義

```
#define LPC_APB0_Base     (0x40000000UL)
#define LPC_ADC_Base      (LPC_APB0_Base + 0x1C000)
```

リスト3　ポインタLPC_ADCの定義

```
#define LPC_ADC
              ( (LPC_ADC_TypeDef  *) LPC_ADC_Base )
```

LPCXpresso IDE と C 言語による

タイマを使った PWM 波形の出力方法

第9章

桑野 雅彦
Masahiko Kuwano

第 5 章では，Micro BASIC を使ったタイマの操作方法を紹介しました．

本章では，C 言語からのアクセス・サンプルということで，LPCXpresso の環境からアクセスしてみることにします．ここで紹介するのは，LPC1343 の内蔵タイマ CT16B1 を利用して周波数を変えながら方形波出力を行うサンプルです．

基本的なアクセス方法は，Micro BASIC からの場合と違いはありません．タイマの内部構造や使い方，レジスタの詳細については，第 5 章の Micro BASIC による PWM 出力のページを参照してください．ここでは，LPCXpresso の環境から扱うときのポイントに絞って説明します．

PWM 波形出力時のタイマの内部構成

図 1 は，サンプルとして作成した PWM 波形出力プログラム起動時のタイマ CT16B1 の内部構成です．

基本的に，Micro BASIC のときと同じですが，与えられているクロックが 72MHz になっている点が異なります．

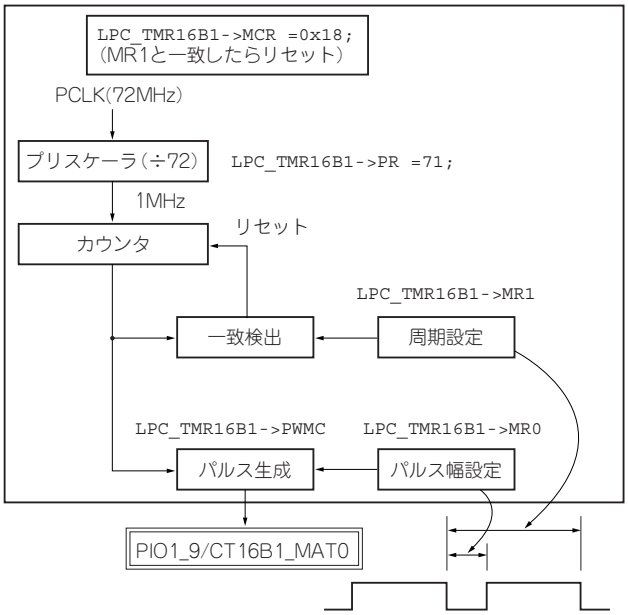


図 1　CT16B1 を使ったパルス出力

GPIO接続したタッチ・スイッチ電極からON/OFF

タッチ・スイッチICの使い方

第10章

桑野 雅彦
Masahiko Kuwano

本章では，静電容量式タッチ・スイッチICの状態読み取りプログラムの作り方を解説します．

読み取った状態は，USB→VCOMでターミナル・ソフトウェアに表示します．

タッチ・スイッチIC PCF8883の特徴としくみ

● タッチ・スイッチIC PCF8883の特徴

付属基板に搭載されている静電容量式タッチ・スイッチICはNXPセミコンダクターズのPCF8883です．内部ブロックを**図1**に，ピン配置を**図2**に示します．

タッチ・スイッチICとマイコンはGPIOで接続します．PCF8883の特徴は次の通りです．

- ダイナミック近接スイッチ
- 感度調節可能
- 応答時間調節可能
- 入力容量値範囲： 10p～60pF
- 自動校正
- パッドとIC間を数メートルにすることが可能
- オープン・ドレイン出力(Pチャネル MOSFET)
- 押しボタン，トグル，パルス出力のどれかに設定可能
- 動作電圧範囲： V_{DD}=3～9V
- 電圧レギュレータ内蔵

● PCF8883のしくみ

静電容量式タッチ・スイッチは，指とセンサ電極の間の静電容量の変化を検出して，電極へのタッチ(ON/OFF)を検出するものです．

付属基板の回路は**図3**のようになっています．図で点線になっているのは，付属基板では実装されていない部品を示しています．

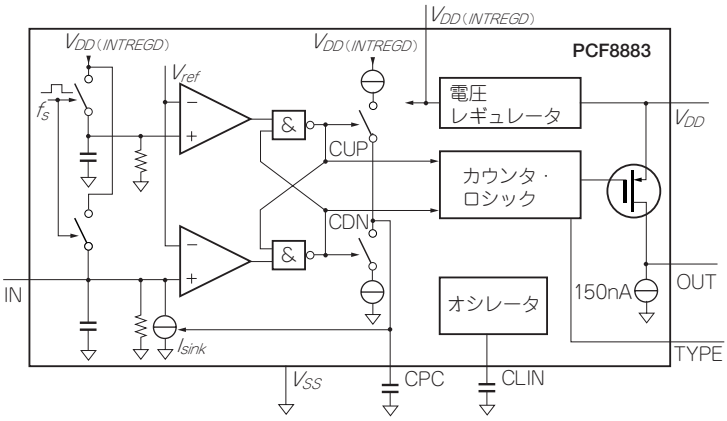


図1　タッチ・スイッチIC PCF8883の内部ブロック

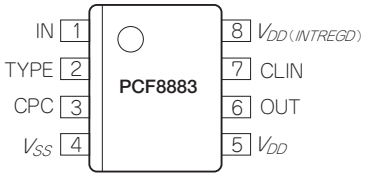


図2　PCF8883のピン配置