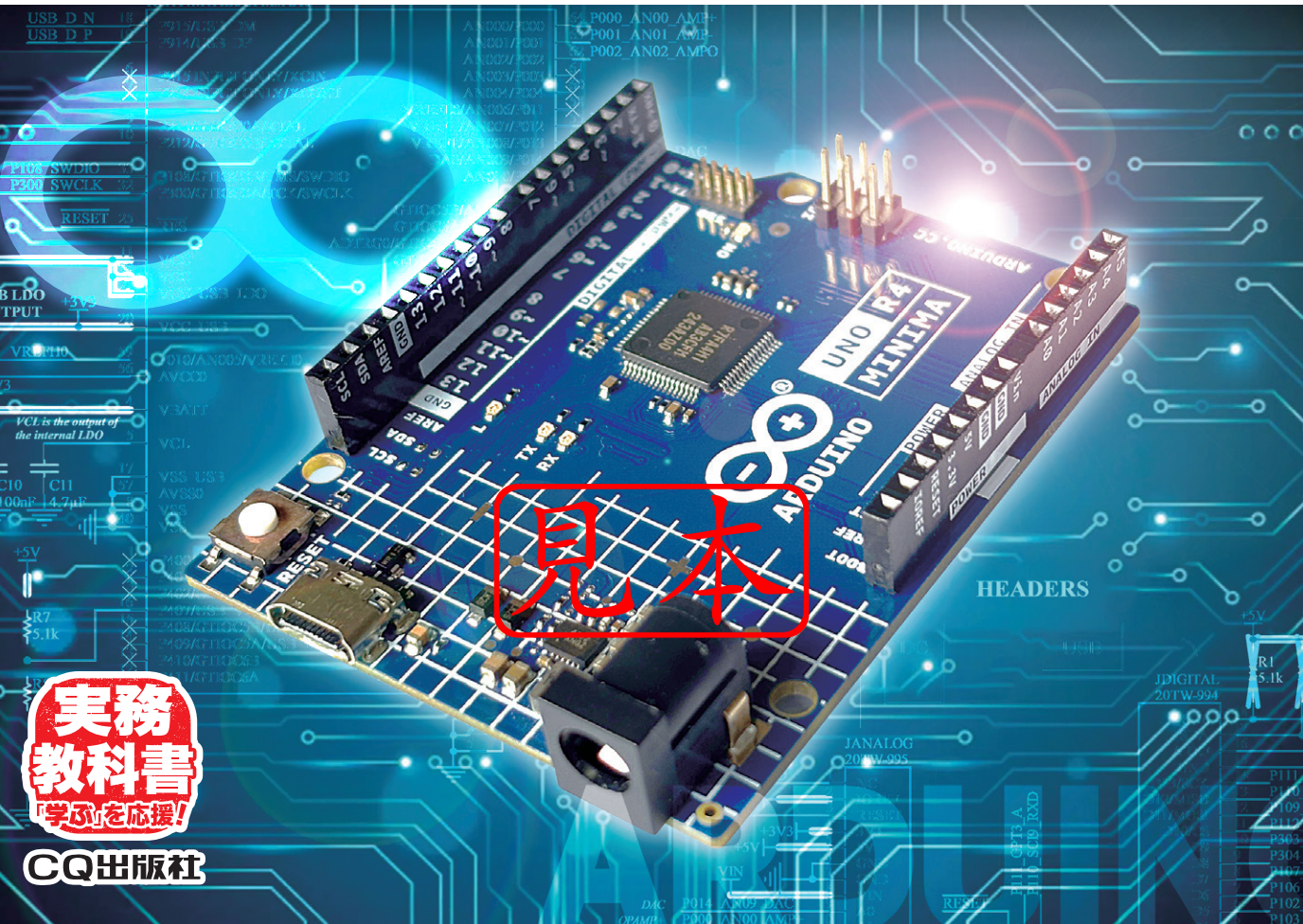
研究や実務に役立つエレクトロニクスの参考書

2026
Spring
No.174

トランジスタ技術 SPECIAL

制御プログラムはなぜ動くのか？ Arduinoがわかれば他のマイコンもOK!

絵とき教科書 Arduino流 マイコンのしくみ&動かし方



実務教科書
「学ぶ」を応援!

CQ出版社

第1章 書き込み・制御で重要「周辺機能」・操作の基本「レジスタ」

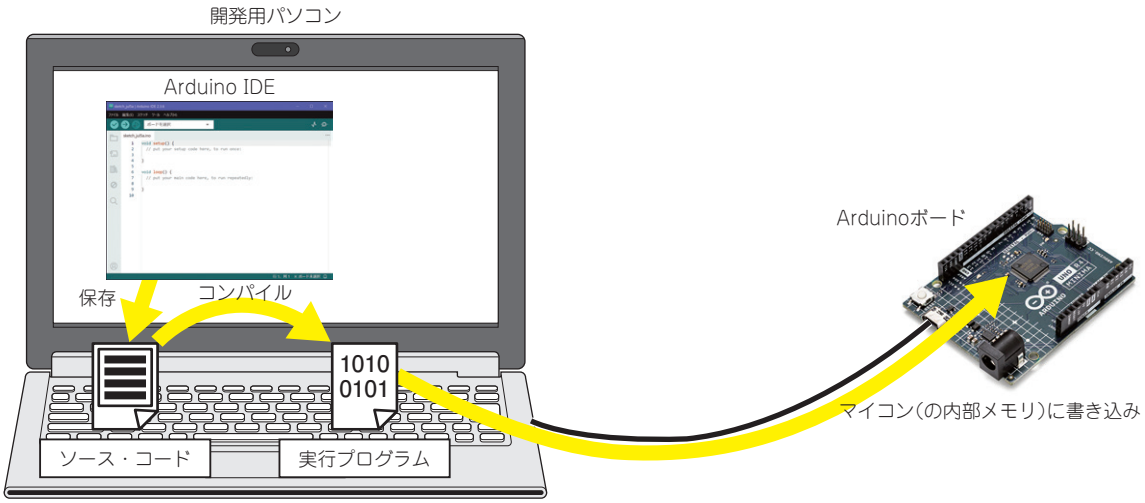
マイコン上でプログラムを動かす基本的なしくみ

山本 秀樹 Hideki Yamamoto

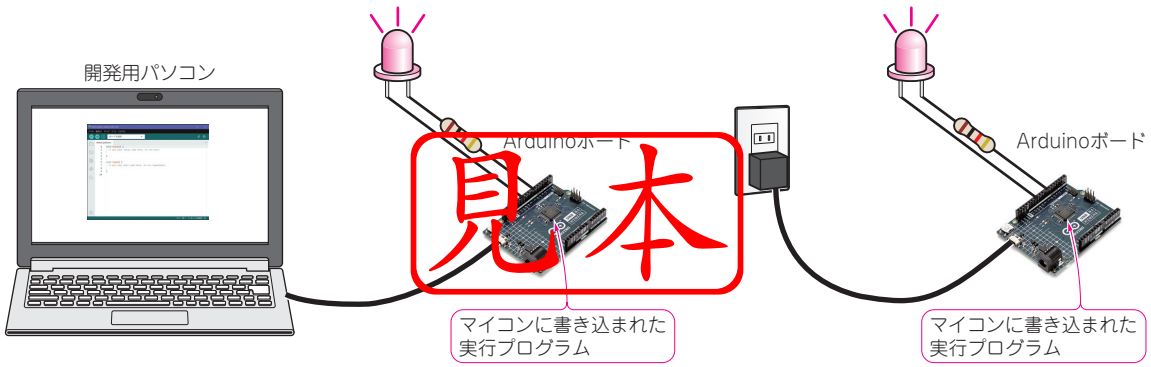
本章ではマイコン内がどのような構成になっているのか，構成要素がどのように動作するのかについて，ざっくり説明します．この内容が次章以降の説明の基礎となるので，しっかり理解するようにしてください．

作成したプログラムをマイコン上で動かすしくみ

開発用パソコンでプログラムを作成し，Arduinoボードに搭載されたマイコンに書き込むまでの流れを**図1**(a)に示します．



(a) 開発環境Arduino IDE上でプログラムの作成＆コンパイルから実行プログラムへのマイコンへの書き込みまですべて行える



(b) マイコンへの書き込みが完了するとプログラムが動作する

(c) 開発用パソコンを外しても書き込まれたプログラムは動作する

図1 作成したプログラムをマイコン上で動かすしくみ

● ソース・コードの作成

プログラムの作成には，プログラム開発ソフトウェア「Arduino IDE」を使います．

Arduino IDE上で，**リスト1**のようなプログラム(Arduino IDEではスケッチと呼ぶ)を記述します．プログラムを保存すると，開発用パソコン内にファイルとして保存されます．このファイルをソース・コードと呼びます．

ソース・コードは人間が読める内容になっています．

● コンパイル…実行プログラムの作成

Arduino IDEでコンパイル(「検証」または「書き込み」)を行うと，ソース・コードから実行プログラムと呼ばれるファイルが開発用パソコン内で作成されます．

ここで注意してほしいのは，作成した実行プログラムはマイコンで実行できるものであることです．基本的に，開発用パソコンでは実行できません．

● マイコンへの書き込み

Arduino IDEで書き込みを行うと，作成した実行プログラムをマイコン(の内部メモリ)に書き込みます．Arduino IDEの「書き込み」は，コンパイルとマイコンへの書き込みをまとめて行います．

● マイコンでの実行

マイコンに書き込んだ実行プログラムは開発用パソコンから見ると内容がわからないものでしたが，マイコンから見ると，理解できる(実行できる)プログラムです．Arduinoの場合は書き込みが終わると，そのプログラムを先頭から実行することになります[**図1**(**b**)]．

リスト1　Arduino IDEに記述するプログラムの基本構成…1回だけ呼び出すsetup関数と繰り返し呼び出すloop関数がある
ArduinoのD13端子につながっているLEDを点滅させる

```
void setup() {
  pinMode(13, OUTPUT); // D13ピンを出力モードに設定
}

void loop() {
  digitalWrite(13, HIGH);   // D13ピンに"H"を出力
  delay(200);               // 200ms待つ
  digitalWrite(13, LOW);    // D13ピンに"L"を出力
  delay(200);               // 200ms待つ
}
```

● マイコンに書き込まれたプログラムは開発用パソコンがなくても動作する

プログラムをマイコンに書き込むと，マイコンを開発用パソコンから切り離しても書き込まれたプログラムは残っていて，電源をつなぐとマイコンはそのプログラムを実行します［**図1**(**c**)］．

マイコン内部の3大構成要素

マイコン内部はいろいろな要素からできています．その中から思い切って3つの重要な構成要素を挙げると，**図2**に示すようにメモリ，CPU(Central Processing Unit)，周辺機能と言えます．

● メモリ

メモリはプログラムやデータを一時的または永続的に保持するものです．**図1**で実行プログラムをマイコンに書き込みましたが，その書き込み先はメモリです．

メモリは大きく分けて，電源を切っても情報を永続

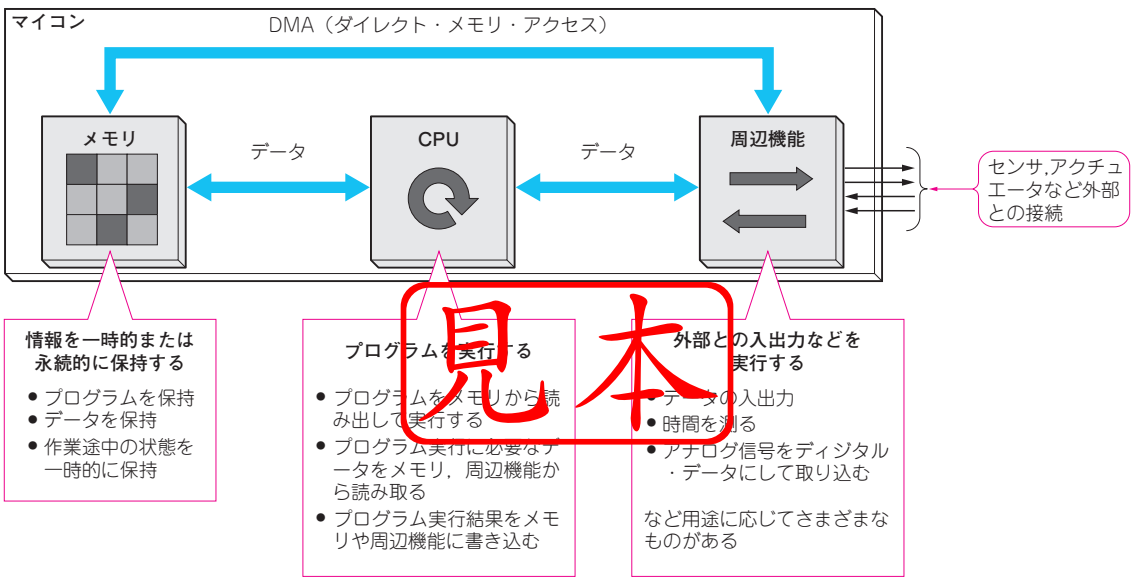


図2　マイコン内部の3大構成要素…用途に応じて多数の「周辺機能」をもつマイコンが一般的

マイコン・プログラムの処理の流れ

山本 秀樹 Hideki Yamamoto

Arduinoのプログラムには，パソコンやスマホのアプリ（アプリケーション・プログラム）とは異なる特徴があります．これは機器に組み込まれるマイコンのプログラム（組み込みプログラム）の一般的な特徴でもあります．

マイコン・プログラムは基本的に終了しない

パソコンやスマホのアプリ（例えばArduino IDEなど）であれば，**図1**(a)のように終了することができます．また終了した後でも，何らかの操作を行えば，そのアプリを再び実行することができます．

● マイコン・プログラムが終了するとどうなるか

一方，**図1**(b)のような空気清浄機の組み込みプログラムを考えてみます．この機器には押しボタン式の電源スイッチがあり，電源をOFFにできます．

もし電源OFFでマイコン・プログラムを終了できたとすると，どうなるでしょうか．再び電源ボタンを押して電源をONにする操作を行っても，そのボタン操作を読み取るプログラムが動作していないので，その機器は電源をONにできなくなります．

このように，機器に組み込まれるプログラムは終了すると問題が起こるので，基本的には終了しません．

プログラムの処理の流れ

● パソコンやスマホのアプリの場合

パソコンやスマホのアプリの処理の流れを極端に簡略化してフローチャートにしたのが**図2**(a)です．「はじまり」から「終わり」までの流れになります．終了操作を行うと「終わり」に進みます．

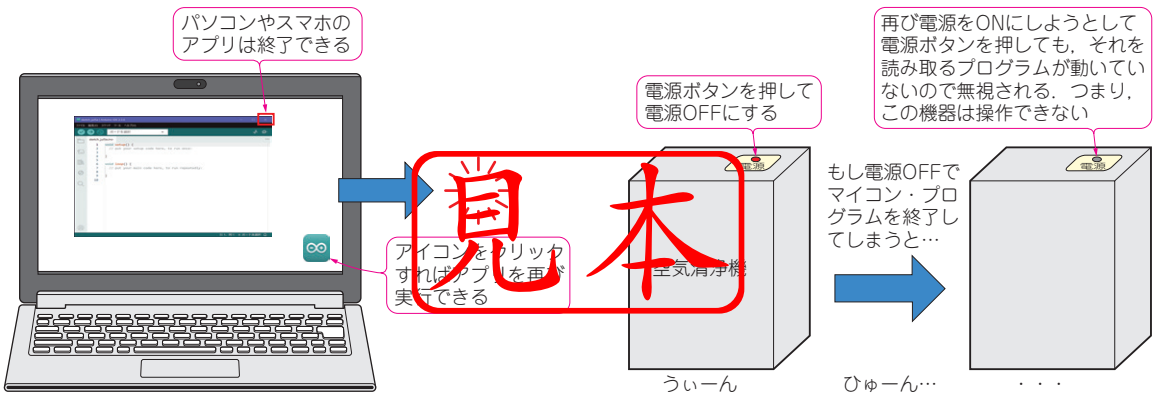
● マイコン・プログラムには終わりがない

一方，マイコン・プログラムでは基本的に，**図2**(b)のように「終わり」がありません．

もう少し細かく見てみましょう．**図1**(b)のような機器のマイコン・プログラムについて，電源スイッチに着目した処理の流れを**図2**(c)に示します．

▶機器の電源がONの場合

図2(c)「はじまり」から開始して，電源がONであれば，電源ONのときに行う処理を繰り返します．



（a）パソコンやスマホのアプリは終了できる　（b）マイコン・プログラムは終了できない

図1　マイコン・プログラムは終了しない
機器に組み込まれたマイコン・プログラムは，終了してしまうと，そのプログラムが組み込まれた機器を操作できなくなる

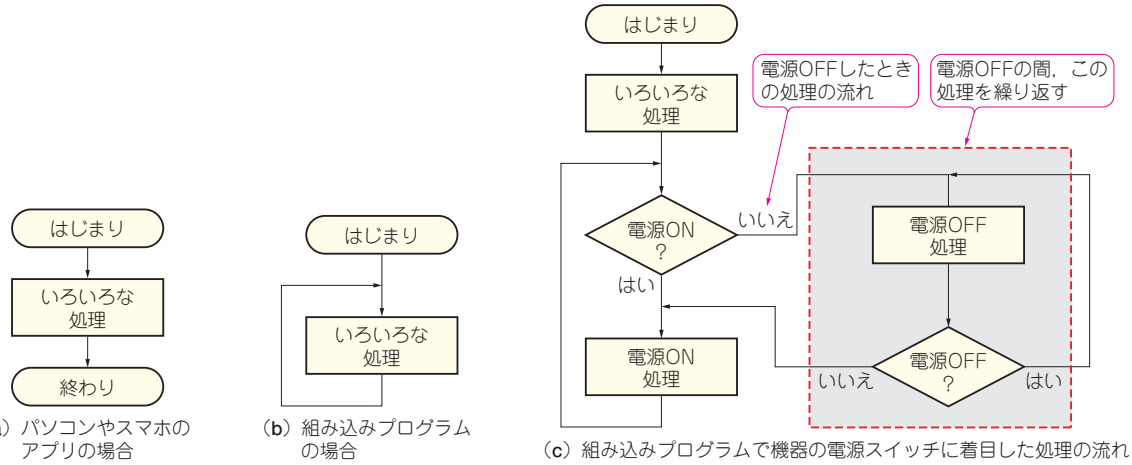


図2 プログラムの処理の流れ
機器の電源をOFFしてもプログラムは終了せず，電源OFFの処理を続ける

▶機器の電源がOFFの場合

ここで電源OFFの操作が行われると，「電源ON？」の条件分岐から「いいえ」に進んで，電源OFFの処理のほうに進みます．ここで電源OFFの処理を繰り返します．電源ONの操作が行われると，「電源OFF？」の条件分岐から「いいえ」に進んで，電源ONの処理のほうに進みます．

このように，組み込みプログラムでは電源OFFであってもプログラムを終了せず，何らかの処理を繰り返します．

1回だけ行う処理setup関数と繰り返し行う処理loop関数

● 処理の流れをもう少し具体化してみる

組み込みプログラムの処理の流れの基本は**図2**(b)のとおりですが，これをもう少し具体化すると，よくあるパターンとして**図3**(a)のような流れがあります．

プログラムの中には，初期化のように1回だけ行えばよい処理と，繰り返し実行する必要がある処理があります．それをこの図のように分けて考えればすっきりします．

● Arduinoのプログラムの構造

Arduinoでも，処理の流れはこの考え方を採用しています．Arduinoの場合はこの考え方に基づき，**図3**(b)のようなプログラムの構造になっています．

Arduino IDEで新規スケッチ(プログラム)を作ると，中が空のsetup関数とloop関数が用意されます(**図4**)．main関数はありません．ユーザはこのsetup関数とloop関数の中身を作成することになります．

実は，main関数はユーザの目に見えないところでArduino IDEにより用意されていますが，ユーザが手を加えることはできません．main関数からは一度だけsetup関数が呼び出され，loop関数は繰り返し

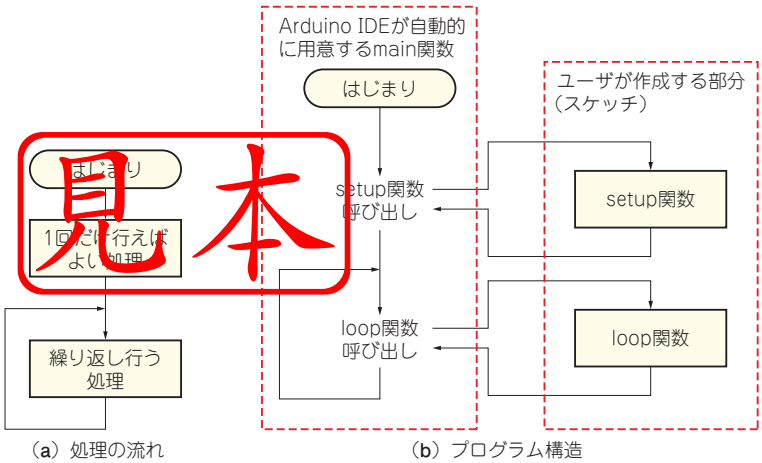


図3 Arduinoプログラムの処理の流れとプログラム構造

第3章 “L”を出力するときは電流が流れ込む…内部回路もふまえることが重要

絵ときI/O制御①… 出力ポートで外部機器をON/OFF

山本 秀樹 Hideki Yamamoto

本章では，最も基本的な周辺機能の1つである出力ポートについて説明します．LEDを点灯/消灯するのも，出力ポートを使って実現しています．

なお，出力ポートや入力ポートは，まとめてGPIO（General Purpose Input/Output，汎用入出力）と呼ばれることもあります．

マイコンの出力ポートから外部LEDを点灯/消灯するしくみ

● LEDを点灯/消灯する回路

基本的な事柄として，LEDを点灯するにはどうするかを説明します．

LEDは電流をアノードからカソードの向きに流すと点灯します．電源とLEDをつなぐのですが，それだけでは電流が流れすぎてLEDが壊れるので，**図1**(**a**)のように電流を制限する抵抗を入れます．LEDの点灯/消灯をマイコンで制御するには，LEDに電流を流すかどうかを制御すればよいことになります．

● 電源まわりの記号はわかりにくい

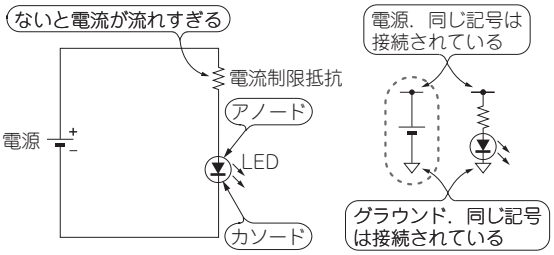
LEDに限った話ではありませんが，回路図では電源がわかりにくく書かれることも多いので，説明します．**図1**(**b**)の抵抗の上にあるT字形の記号や，LEDの下にある三角の記号は，同じ記号が接続されていることを示します．**図1**(**a**)と**図1**(**b**)は見た目は異なりますが，同じ回路を示します．さらに，破線で囲っている電源のマークが省略される場合もあります．

● マイコンによるLEDの点灯/消灯の考え方

図2は，LEDのアノードを抵抗を介してマイコンの出力ポートにつなぎ，LEDのカソードは電源のマイナス側につないでいます．**図2**(**a**)のように，アノードが電源のプラス側につながるようにマイコン内部を制御できれば，LEDに電流が流れて点灯します．

逆に**図2**(**b**)のように，アノードが電源のマイナス側につながるようにマイコン内部を制御できれば，LEDの両端が電源のマイナス側につながることになるので電流が流れず，LEDは消灯します．

このように，マイコンが電源のプラス側の電圧を出力することを，「マイコンが“H(High)”レベルを出力する」と言うことにします．同様に，マイコンが電源のマイナス側の電圧を出力することを，「マイコン



(**a**) 向きを間違えないように接続する必要がある　(**b**) 電源はわかりにくく書かれていることも多い

図1 LEDを点灯する基本回路…抵抗を介して電源をつなぐ
LEDはアノードからカソードに電流を流すと点灯する

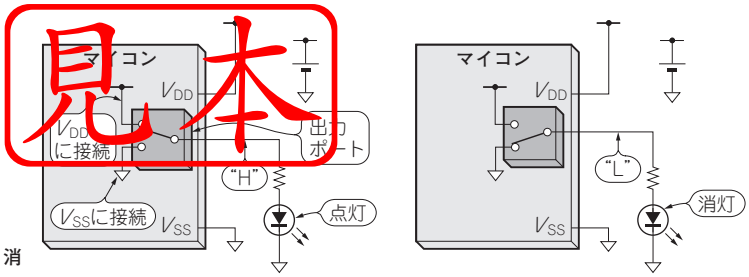
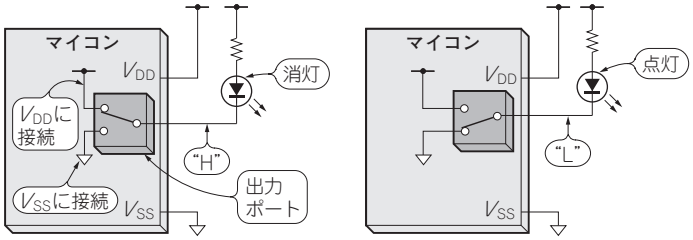


図2 マイコンの出力ポートからLEDを点灯・消灯するときの回路動作
出力ポートは電源につながる(“H”)かGNDにつながる(“L”)

(**a**) マイコンが“H”レベルの電圧を出力するとLEDは点灯　(**b**) マイコンが“L”レベルの電圧を出力するとLEDは消灯



(a) マイコンが"H"レベルの電圧を出力するとLEDは点灯　(b) マイコンが"L"レベルの電圧を出力するとLEDは消灯

図3　LEDのつなぎ方で点灯/消灯は変わる

が"L(Low)"レベルを出力する」と呼ぶことにします.

● 出力ポートのイメージは内部切り替えスイッチ

マイコン内にスイッチを設けて，そのスイッチから"H"レベル(電源のプラス側)の電圧を出すか，"L"レベル(電源のマイナス側)の電圧を出すか選択できれば，マイコンによるLEDの点灯/消灯は実現できそうです.

このスイッチから"H"の電圧を出すか"L"の電圧を出すか制御できるものが，周辺機能の1つである出力ポートです.

● LEDのつなぎ方はほかにもある

LEDとマイコンのつなぎ方には，**図2**のようにアノードをマイコンの出力ポートにつなぐ方法のほかに，**図3**のようにカソードをマイコンの出力ポートにつなぐ方法もあります.

図3のようにすると，**図2**とは逆に，マイコンが"H"を出力するとLEDの両端が電源のプラス側に接続されたことになり，LEDは消灯します．また"L"を出力すると，電源からLEDを通ってマイコンに電流が流れ込み，LEDが点灯します.

このように，LEDを点灯するためにマイコンが"H"と"L"のどちらを出力すればよいかは，LEDのつなぎ方によって決まります.

● 出力ポートでも電流が流れ出すとは限らない

図3(b)のように出力ポートから"L"を出力した場合，出力しているのにマイコンに電流が流れ込むことになり，「出力」という表現に違和感を感じるかもしれません．ここでいう出力は，マイコンが電圧を決めることを意味しており，電流の流れる向きについては何も言っていません．マイコンが電圧を出力したとしても，マイコンから電流が流れ出す場合も，流れ込む場合も，電流が流れない場合もあります.

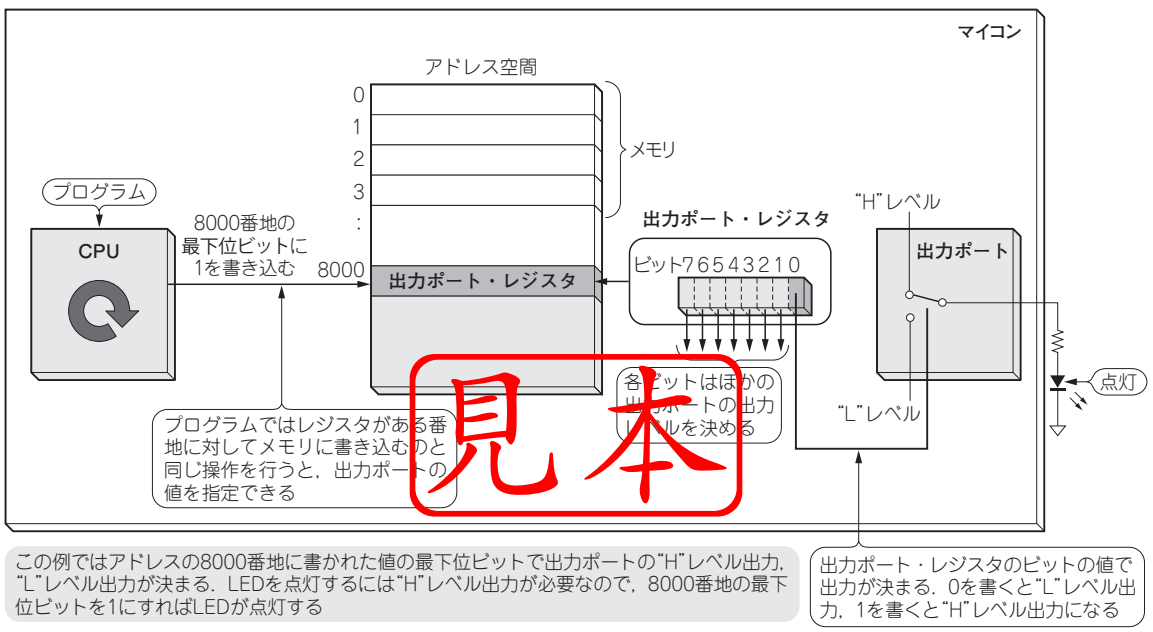


図4　特定のレジスタに値を書き込むことで出力ポート(周辺機能)を制御できる
どのレジスタに値を書き込めばよいかはマイコンのマニュアルに記述されている

第4章　必ず方向レジスタを設定しないといけない理由

絵ときI/O制御②…入力ポートで外部信号を取り込む

山本 秀樹 Hideki Yamamoto

本章では，外部の信号をマイコン内に取り込むための入力ポートについて説明します．また入力ポートは，出力ポートと組み合わせた入出力ポートになっていることが多いので，それも説明します．

外部からの信号をマイコンの入力ポートから取り込むしくみ

入力ポートは，外部からの信号をマイコン内で扱えるディジタル値に変換します．

● 入力ポートは電圧の“H”/“L”を1/0に変換する

出力ポートでは，CPUがレジスタに書いた1ビットの値をもとに，“H”や“L”の電圧として出力していました．入力ポートは逆に，外部からの信号の電圧が“H”か“L”かを判断して，CPUが扱える1ビットの値に変換します．

なおここで説明する入力ポートはあくまでもディジタル入力なので，電圧の高中低など，3つ以上の状態を読み取ることはできません．

入力ポートのイメージを**図1**に示します．

● 入力電圧と入力値の関係

入力値が1や0になるのはどの程度の電圧でしょうか．実は，これはマイコンによって異なります．

本書で実験に用いたArduinoボードのマイコンの場合は，4V以上で1になります（電源のプラス側の電圧が5Vの場合）(1)．他の種類のマイコンでは2V以上で1になるものもあるので，マイコンの仕様書で確認する必要があります．

「入力電圧が4V以上で入力値が1になる」マイコンの場合でも，「4V未満なら0になる」とは限りません．実験に用いているArduinoボードのマイコンの場合，1V以下なら入力値が0になります(1)．こちらもマイコンの種類によって，0.8V以下なら入力値が0になるものなど，さまざまです．

その間の電圧が入力されたときは，入力値が0になるか1になるかはわかりません（**図2**）．そういう中途半端な電圧が入力されることは想定していないのです．はっきりした“H”または“L”が入力されることを想定しています．

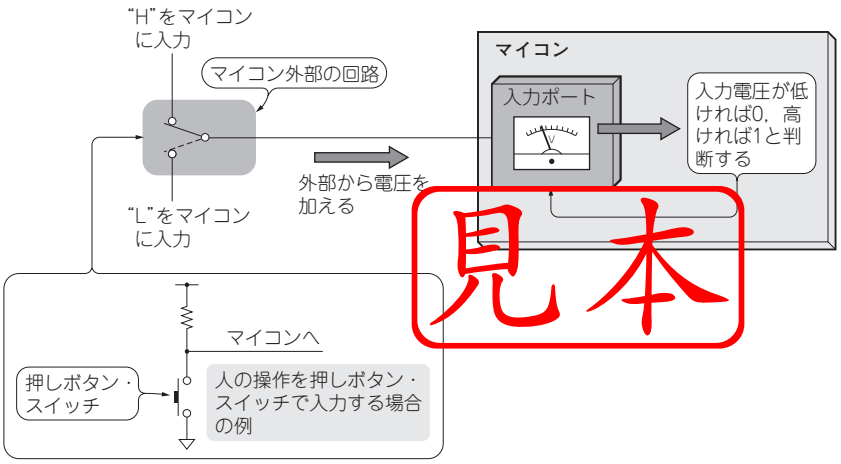


図1　入力ポートのイメージ
入力ポートは外部からの信号をマイコン内で扱えるディジタル値に変換する

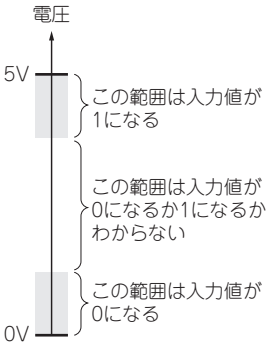


図2　入力電圧と入力値の関係

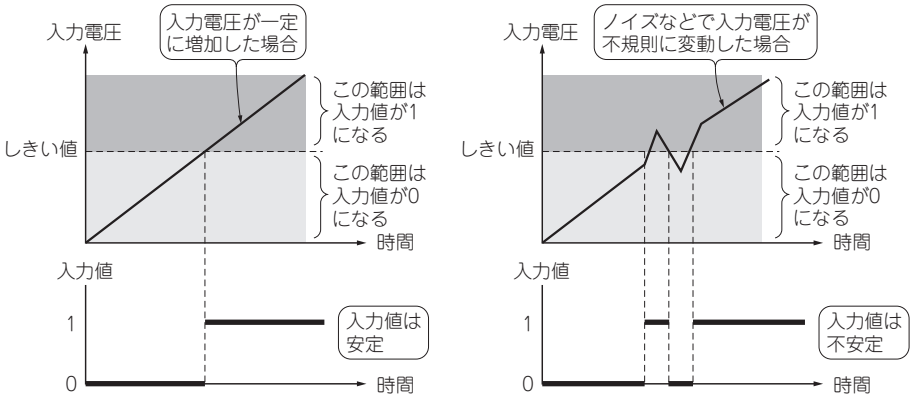


図3　微妙に変動する入力電圧を読み取るくふう…しきい値が1つだと入力値が不安定になる場合がある

（a）理想的な変化であればしきい値が1つでも問題ない

（b）しきい値付近で入力電圧が微妙に変動すると入力値が不安定になる

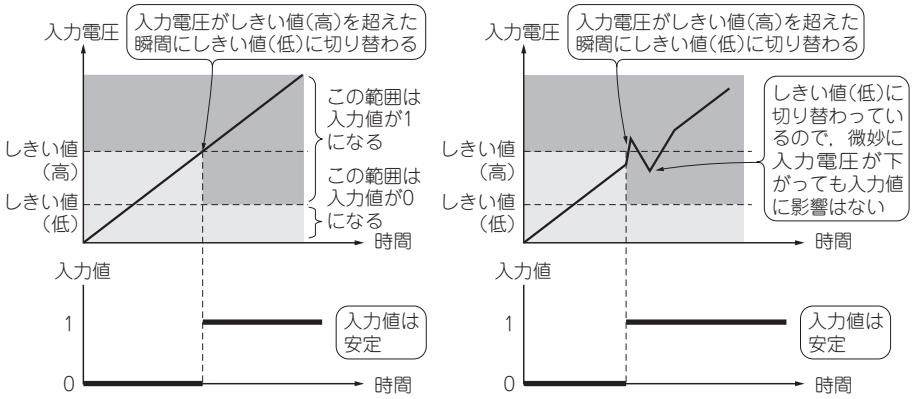


図4　2つのしきい値を切り替えて使うと入力値が安定する（シュミット・トリガ入力）

（a）入力電圧が緩やかに増加した場合

（b）しきい値が切り替わるので入力電圧が微妙に変動した場合も入力値に影響しない

● 中途半端な入力電圧に対応するにはシュミット・トリガ入力を使う

入力ポートによっては，中途半端な電圧に対応したものもあります．ただし，例えば「入力電圧が2.5 Vより高ければ入力値は1，低ければ入力値は0」のように単純に区切ると，問題が起こる可能性があります．

▶しきい値が1つの場合

しきい値を1つだけ設定すると，**図3**(a)のように入力電圧が一定に増加する場合は，しきい値を超えたところで入力値が変化します．

しかし，しきい値付近でノイズなどにより入力電圧がわずかでも変動すると，**図3**(b)のように不安定な入力値になってしまいます．

▶2つのしきい値を切り替える

図4(a)のように，2つのしきい値を設定して切り替える方法があります．入力電圧が低いときは高い方のしきい値で判断し，入力電圧がしきい値より低い間は入力値は0になります．

入力電圧が上がり，高い方のしきい値を超えると，入力値は1になり，同時に今度はしきい値を低い方に切り替えます．入力電圧がしきい値より高い間は，入力値は1になります．

この方法だと，しきい値付近で入力電圧が変動しても，入力値が不安定になることはありません［**図4**(b)］．入力電圧が下がっていく場合も同様です．このような入力方法を，シュミット・トリガ入力と呼びます．

外部の信号は入力ポート・レジスタで読み取れる

入力ポートも周辺機能の一種なので，CPUはレジスタを経由して入力電圧に対応する入力値を読み取ることになります．

図5では8010番地に入力ポート・レジスタを割り当てています．このレジスタの最下位ビットに入力ポートがつながっています．CPUは，メモリを読み取るのと同じようにこのレジスタを読み取ると，その最下位ビットが外部からの入力になっています．

図5にはありませんが，入力ポート・レジスタの他のビットは，他の入力ポートにつながっています．

第5章 使いまくる周辺機能「タイマ」入門

時間を決めて「待つ」…絵とき「タイマ」処理

山本 秀樹 Hideki Yamamoto

本章では，マイコンの周辺機能の1つであるタイマを使って，プログラムの中で時間を決めて待つ方法について説明します．

ここまでに扱ってきたArduinoのプログラムを見ると，delay関数で時間待ちを行っています．delay関数呼び出しを行ったとき，マイコンの内部ではどのようなことを行っているのかを説明します．

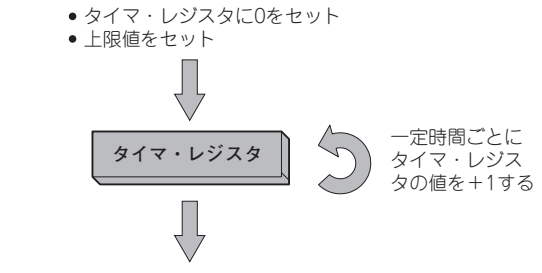
タイマの動作イメージ

● 一定時間ごとにカウントアップして上限値になれば終了

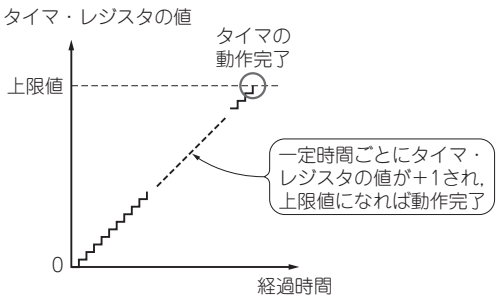
時間を計る方法の1つに，0から一定時間ごとにカウントアップしていって，計りたい時間になれば終了する，という方法があります．今回はそういうタイマを考えてみます．

図1(a)でタイマ・レジスタに0を設定しておいて，一定時間ごとに値を1ずつ増やしていきます．このレジスタの値が上限値(計りたい時間の値)になれば終了です．例えば計りたい時間が1秒(1 s)で，1 msごとにタイマ・レジスタの値を増やすのであれば，上限値は1000です．

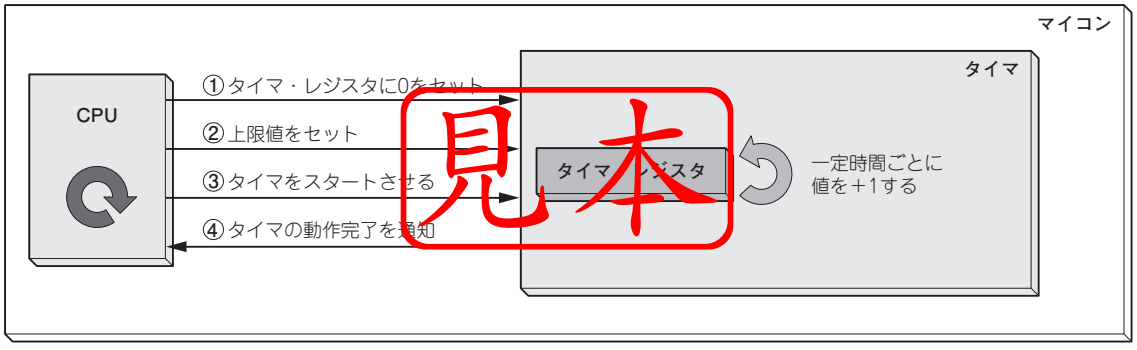
タイマ・レジスタの値の変化を**図1**(b)に，CPUとタイマの関係を**図1**(c)に示します．



(a) タイマ・レジスタの値を増やしていく



(b) タイマ・レジスタ値の変化



(c) CPUとタイマのやりとり

図1　時間を計る…原理的なタイマの動作

第8章 PWM出力analogWrite関数の動作と使い方

絵ときPWM出力… アナログ値制御を実現する

山本 秀樹 Hideki Yamamoto

マイコンは基本的に0か1のディジタル値を扱いますが，疑似的にアナログ値を出力する機能として，PWM(Pulse Width Modulation；パルス幅変調)出力機能があります．PWM出力機能は，疑似的なアナログ出力以外にも，さまざまな目的で用いられます．

本章では，PWM出力機能を使った疑似的なアナログ出力について解説します．

PWM出力によるディジタル信号アナログ出力とは

● 0と1を高速に切り替えて中間値に見せる

第2章で解説した，ディジタル出力(出力ポートに0か1を出力する)によってLEDを点灯/消灯するプログラムでは，LEDの状態は「点灯」か「消灯」しかなく，20％くらいの明るさでぼんやり点灯する，といったことができませんでした．

しかし，もし点灯と消灯を高速に切り替えることができれば，点灯する割合を20％，消灯する割合を80％にして高速に切り替えれば，人間の目には疑似的に20％の明るさに見えそうです．

● PWM出力の動作イメージ

PWM出力は，1と0を出力する時間の比率を変えてディジタル出力を行う機能です．この機能を応用すると，疑似的なアナログ信号を出力できます(**図1**)．

点灯する時間と消灯する時間の比率を同じにすると中間的な明るさになり［**図1**(a)］，点灯する時間を長くするとやや明るく［**図1**(b)］，点灯する時間を短くするとやや暗く［**図1**(c)］なります．

PWM出力機能の実現方法

PWM出力を行うとき，マイコン内でどのように動作するのか説明します．

● PWM出力機能の基本構成…タイマを使う

PWM出力機能は，時間を計る機能であるタイマをもとに，少し機能を付け足して実現できます．

タイマの値は，0から一定時間ごとに1ずつ増えていき，上限値に達すると0に戻って再度1ずつ増えて

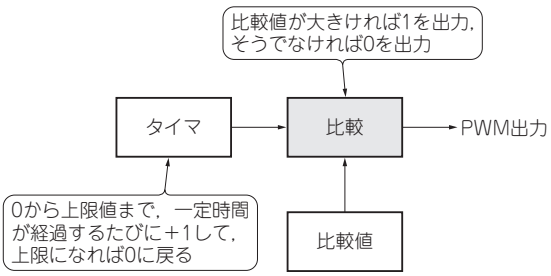
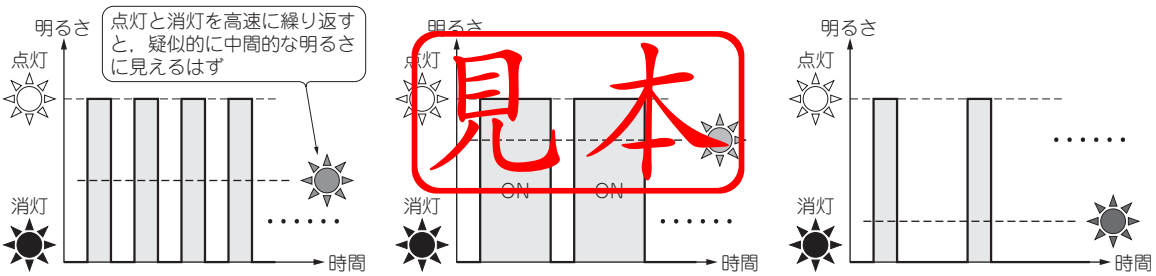


図2 PWM出力機能の基本構成



(a) ディジタル値を高速に変化させて，疑似的に中間値を作る

(b) 点灯する比率を高くすると，明るくなる

(b) 点灯する比率を低くすると，暗くなる

図1 PWM出力によるアナログ値制御のイメージ

第15章 超音波距離センサなど…パルス幅を測りたいことは意外とある

入力パルス幅を測る

山本 秀樹 Hideki Yamamoto

センサにはさまざまな出力のものがあります．ここでは，パルス幅で信号を出力する超音波センサを題材に，入力パルスの幅をマイコンでどのように読み取るかについて解説します．

入力パルスの幅を測りたい場合がある

ここまで，マイコンへの入力について何通りか取り上げてきました．センサからマイコンへ入力する信号の種類とマイコンへの入力手段を**表1**にまとめます．

スイッチのような単なる“H”/“L”のディジタル信号は入力ポートで読み取れます．可変抵抗の値を読み取りたい場合は，アナログ電圧をA-D変換すれば読み取れます．通信による入力は，それぞれの通信仕様に従って読み取れます．

本章では，まだ解説していなかった入力パルスの幅の測り方について，どのように読み取るかを解説します．

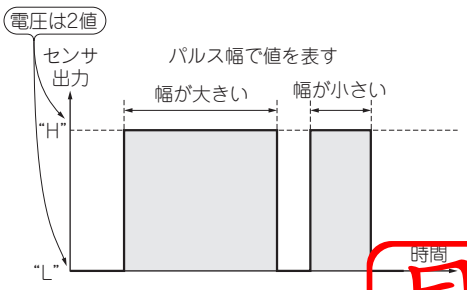


図1 パルス幅で値を表す方法

パルス幅で出力するとは

「パルス幅で出力する」とは，どういう意味でしょうか．**図1**に例を示します．

このセンサは“H”または“L”のディジタル出力を行いますが，値はパルス幅(時間)で表しています．幅が広いと値が大きくなり，幅が狭いと値は小さくなります．

図では“H”のパルスになっていますが，“H”と“L”が反転して，“L”のパルスの場合もあります．

超音波距離センサの例

パルス幅で出力するセンサの例として，超音波距離センサを使います．どのように測定しているのか理解すれば，なぜパルス幅で出力するのか納得できるでしょう．

● 超音波距離センサの構成

超音波距離センサは**図2(a)**のような外観をしています．銀色の2つの円筒は，片方が超音波送信器で，もう片方が超音波受信器です．超音波送信器から超音波を送り，反射してきた超音波を超音波受信器で受信します．

● 時間を計ることで距離を測る

距離を測るときは，超音波送信器から超音波を送ったときに“H”を出力し，反射してきた超音波を超音波受信器で受信したときに“L”を出力します［**図2(b)**］．

表1 入力する信号の種類とマイコンへの入力手段

センサからマイコンへ入力する信号の種類	マイコンへの入力手段	センサ例	本書での解説
ディジタル値(0/1)を入力	入力ポート	スイッチ	4章
抵抗値，電圧，電流などのアナログ値を入力	電圧に変換してA-D変換	可変抵抗	9章
ディジタル値をパルス幅で入力	タイマ	超音波距離センサ	15章(本章)
ディジタル値をパルス数で入力	カウンタ	回転数センサ	16章
通信で入力	通信インターフェース	温湿度センサ	10章

第20章 同じプログラムをいろんなマイコン・ボードに書き込んで使える

マイコン・プログラミング超定番環境 Arduino IDEのしくみ

山本 秀樹 Hideki Yamamoto

本章では，Arduinoボードのプログラム開発に使うソフトウェア「Arduino IDE」について解説します．複数のマイコン・ボードに対応できるしくみから，入手方法，インストール方法，設定までを解説します．

マイコン・プログラミング超定番環境Arduino IDEの特徴

● プログラミングからマイコン・ボードへの書き込みまで行える

Arduino IDEは，Arduinoボードと組み合わせて使用するプログラム開発ソフトウェア(IDE；Integrated Development Environment，統合開発環境)です．これを使うと，Arduinoボードで動くプログラムを作成でき，ボードへの書き込みや実行も簡単な操作で行えます．

Arduino IDEは，Arduinoの公式サイトから無償でダウンロードできます．

● 1つのArduino IDEがいろんな種類のArduinoボードに対応できる

Arduino IDEは，1つで多数のArduinoボードに対応しています．そのしくみを説明します．

Arduino IDEはArduinoボードに依存しない共通部分と，個々のArduinoボード専用の個別部分からなります．この共通部分と適切な個別部分を組み合わせることで，Arduino IDEはArduinoボードの開発環境として動作します(**図1**)．

● Arduinoボード以外でもArduino IDEを使える

Arduinoボード以外のマイコン・ボードでも，そのボード用の個別部分のデータが用意されていれば，Arduino IDEを使ってプログラムを開発できます．

これによりArduino IDEは，さまざまなマイコン・ボードで使えるようになっています．

Arduino IDEの入手

● Arduino IDEのダウンロード

Arduino公式ホームページからArduino IDEをダウンロードします．

https://www.arduino.cc/

画面上部のメニューから［Products］-［Arduino IDE］を選択すると，Arduino IDEのページが表示されます(**図2**)．執筆時点ではバージョンが2.3.6でしたが，新しいバージョンが出ていたら最新版を使います．

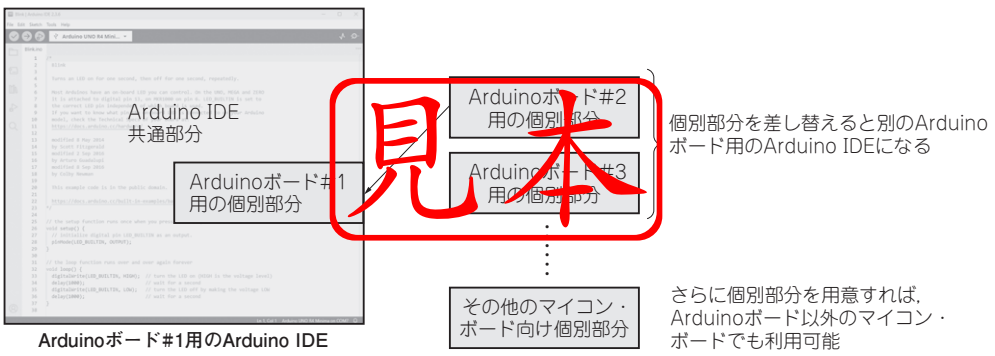


図1 Arduino IDEの共通部分と個別部分(再掲)
Arduino IDE上で，ボードごとに異なる個別部分を差し替えることで，さまざまなArduinoボードに対応できる

付録 Arduinoボード以外のマイコン・ボードも対応！

同じArduino IDEでラズパイPicoプログラミング

白阪 一郎 Ichiro Shirasaka

Arduino IDEは，Arduinoボード用のプログラムを作成するための統合開発環境です．Raspberry Pi Pico 2はArduinoボードではありませんが，Arduino IDE用のRaspberry Pi Pico 2ボード・ライブラリがリリースされているため，手順に従って環境設定を行うことで，Arduino IDE環境を使って簡単にアプリケーション開発を行えるようになります．

ここでは，Arduino IDEを使ったRaspberry Pi Picoのマイコン・プログラミングについて解説します．

1　ラズパイPicoプログラミングの準備

Raspberry Pi Pico 2をArduino IDEでプログラミングするために必要な開発環境の構築手順を解説します．Arduino IDEにPico 2を認識させるための設定や，実際にプログラムを書き込むまでの準備を説明します．

マイコン・ボードRaspberry Pi Pico 2

● 安価でサンプルやライブラリも豊富

Raspberry Pi Pico 2は，デュアルコアのARM Cortex-M33相当のCPUであるRP2350を搭載したマイコン・ボードです．1000円台の価格でありながら，ArduinoやMicroPythonでアプリケーション開発が行え，さまざまなI/Oインターフェースをプログラムで実現できるプログラマブルI/O機能（PIO）などの特徴から，数多くの応用例が発表されています．豊富なサンプル

表1-1　Raspberry Pi Pico 2の各ピンの機能の概要

ピン名	説　明
VBUS	USBからの5 Vが出力される
VSYS	ボードに電源を供給する（1.8 V～5.5 V）．USBホスト動作でUSBから接続機器に5 Vを供給する場合は5 Vを供給する
3V3_EN	GNDレベルにするとボード内回路への電源供給をOFFにする
3V3(OUT)	3.3 Vを外部に出力する
ADC_VREF	ADCの外部リファレンス電圧入力
RUN	GNDレベルにするとプロセッサへの電源供給をOFFにする
ADC0，ADC1，ADC2	3×12ビットADC入力
AGND	アナログGND
GP0～GP22，GP26～GP28	26×GPIO
SPI0，SPI1，SPI2	3×SPI
I2C0，I2C1	2×I²C
UART0，UART1	2×UART

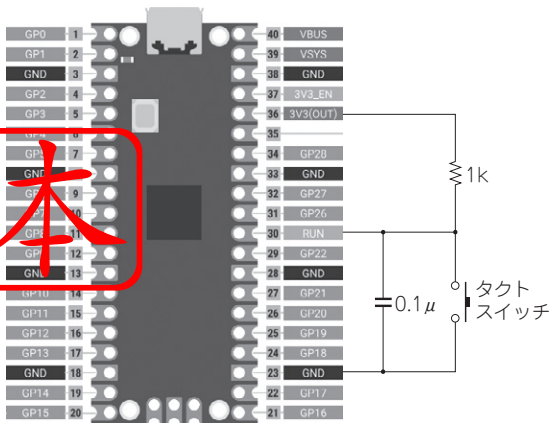


図1-2　リセット・スイッチ回路