

Arduinoの徹底活用

濱崎, 真洋
九州大学応用力学研究所

<https://doi.org/10.15017/7375871>

出版情報 : 九州大学応用力学研究所技術室 技術室報告. 7, pp.40-43, 2025-07. Research Institute for Applied Mechanics, Kyushu University

バージョン :

権利関係 :



Arduino の徹底活用

濱崎 真洋

要 旨

Arduino における基本的な I/O 関数（digitalWrite、digitalRead、analogWrite、analogRead）の動作を分析し、関数によらないポート操作による高速化を図ることで、矩形波出力や信号の高速読み取りや PWM の応用が可能である。筆者がこれまでに作成したプログラムの実例とあわせて、Arduino の潜在的な性能を引き出す方法を示した。新しい機種には及ばないものの、工夫次第で十分に実用に耐えることを確認できた。

キーワード

Arduino 高速化 デジタルノギス ロータリーエンコーダー プラスチック光ファイバー

1. はじめに

Arduino は、ワンボードマイコンの一種であり、センサーなどの外部モジュールとの通信、PC とのシリアル通信などを行うことができる機器である。Arduino 以前のマイコンとは異なり、USB ケーブルで PC に接続し、Arduino IDE でサンプルプログラムを書き込むだけで、「Lチカ」と呼ばれる LED を点灯させる動作テストまで、非常に容易に到達できる。また、Arduino はオープンソースハードウェアであるため、外観や性能の異なる互換機が多数存在し、基本的なプログラムであれば互換性があるため、移植性にも優れている。

筆者も業務で Arduino を使用してきたが、最近では、より高性能で安価な Raspberry Pi Pico などのマイコンを使用することが多くなり、一世代前の Arduino UNO R3 をはじめとする各種 Arduino シリーズの使用頻度は減っているのが現状である。確かに新機種と比べると性能面では見劣りするが、互換性を気にせず性能の限界まで使い込んだ場合、どこまでのことが可能なのか。本稿では、筆者がこれまでに作成したプログラムの紹介を交えながら、その可能性についてまとめる。

2. digitalWrite

digitalWrite は Arduino の基本と言っても過言ではない。LED を光らせるだけでなく、スピーカーを鳴らしたり、I2C、SPI や UART などの通信

規格を自前で実装したりすることもできる。また、小型のモーターであれば駆動することも可能である。そんな digitalWrite だが、Arduino 基板に 16MHz の水晶振動子が搭載されている割に、動作が遅いと感じたことはないだろうか。実際、digitalWrite は内部でさまざまな処理を行った上でピンの電圧を変化させており、図 1 のコードのように動作させた場合、約 85kHz の矩形波が得られた。これは、たとえば UART を実装しようとした場合、115,200bps のボーレートを実現するには厳しい速度である。また、複数ピンを使用したパラレル通信を行う場合、さらに速度の低下が懸念される。そこで、digitalWrite の高速化を検討した。

```
void setup() {  
    pinMode(2, OUTPUT);  
}  
void loop() {  
    digitalWrite(2, HIGH);  
    digitalWrite(2, LOW);  
}
```

図 1 digitalWrite のコード

Arduino は AVR マイコンであり、AVR 向けの C/C++コードが利用可能である。Arduino の統合開発環境である Arduino IDE で記述したスケッ

チは、まずプリコンパイルによって AVR 向けの C/C++コードに変換され、その後 C/C++用のコンパイラにより実行ファイルが生成される。そして、この実行ファイルがシリアル通信を通じて Arduino に書き込まれる。この仕組みにより、AVR マイコンのコードをスケッチ内に記述しても問題なく動作する。

高速化したコードを図 2 に示す。オシロスコープで確認したところ、このコードでは約 4MHz の矩形波（デューティ比 25%）が得られた。ピンアサインについては Arduino 公式 Web サイト^[1]を参照していただきたいが、PORTB、PORTC、PORTD などに 8 ビットの値を書き込むことで、各ピンの出力状態をまとめて制御できる。マイコンが内部で使用しているピンもあるため、特定のピンを HIGH にしたい場合は OR 演算、LOW にしたい場合は AND 演算を用いるのが適切だろう。

```
void setup() {
  for (int i = 0; i <= 7; i++) {
    pinMode(i, OUTPUT);
  }
}
//DDRD = B11111111;
//と書いてもよいが、可読性重視

void loop() {
  while (1) {
    PORTD = B00000100;
    PORTD = B00000000;
  }
}
//whileはloop()よりサイクルが早い
//左からD7～D1ピン
//OR、AND演算の場合は下記
//PORTD = PORTD | B00000100;
//PORTD = PORTD & B11111011;
```

図 2 PORT 出力のコード

ポート出力や演算には 1 クロック、ループジャンプには 2 クロックかかるため、必要に応じて `asm("nop¥n");` などでも 1 クロックの待ちを入れるか、単純なループ `int i = xx`（ループ回数）；`while(i){i--;}` などでも時間調整すれば、任意のパルス幅を生成できる。ただし、これはパルス出力に専念する処理であり、他の処理を並行して行うことができないため、後述の `analogWrite` を使う方

が実用的である。一方、複数ピンを同時に処理できるため、R-2R ラダー型 DA コンバータの構成や、パラレル通信に適している。

3. digitalRead

`digitalRead` は、スイッチの On/Off 検出やロータリーエンコーダーの読み取りなどに用いられる。高速化については、図 3 に示すようなコードで実現可能である。`digitalWrite` と同様、PINB、PINC、PIND には 8 ビット分の各ピンの HIGH/LOW 情報が格納されており、AND 演算を用いて必要なピンの状態を取り出すことができる。`digitalRead` は 1 回あたり約 4.5 μ 秒 (220kHz) かかるが、PIN レジスタを使った読み取りは約 440n 秒 (2.2MHz) で完了する。また、一度に 8 ピン読み取れるメリットもある。

```
void setup() {
  for (int i = 0; i <= 7; i++) {
    pinMode(i, INPUT);
  }
}
//DDRD = B00000000;
//と書いてもよいが、可読性重視

void loop() {
  byte value = (PIND & B00000100) >> 2;
}
// 2ピンを読み、HIGHなら1, LOWなら0が
// 変数valueに格納される
```

図 3 PIN 読み込みのコード

この高速化を応用した事例として、デジタルノギスの出力読み取りがある。筆者が使用しているノギスは電池の消耗が激しく、電池を大型化するために内部配線を確認していたところ、外部出力端子が見つかった。この端子からは、デジタルノギスの出力値が独自フォーマットのデジタル信号で出力されていることが分かった。しかし Arduino でこの信号を読み取ろうとしたところ、`digitalRead` では処理が間に合わなかったため、PIN レジスタを用いた高速な読み取りを行い、出力値の読み取りに成功した。図 4 に、加工したデジタルノギスを示す。余談であるが、ノギスの電源ボタンを押しても液晶表示が消えるだけで、測定および信号出力は継続されていた。これが電池消耗の原因の一つではないかと考えている。

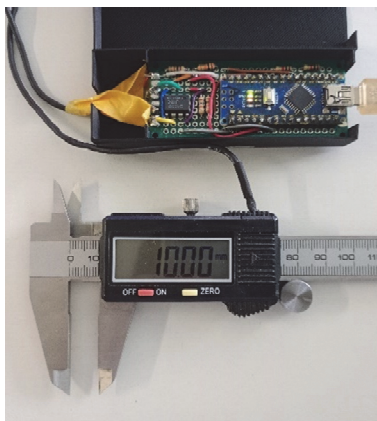


図 4 加工したデジタルノギス

また、ロータリーエンコーダーの読み取りには外部割込み処理を用いて、A 相および B 相の HIGH/LOW を検出する必要がある。外部割込みを 2 ピン使用し、digitalRead にてエンコーダー波形のエッジを検出したところ、1 秒で約 10,000 カウント計測できた。digitalRead の高速化により、回転速度の上限を引き上げることが可能になると思われる。さらに、Arduino Mega など外部割込みピンが複数ある機種では、A 相・B 相をそれぞれ 2 つに分岐し、RISING と FALLING のトリガで割込みをかけることで、ピンの読み取りなしに処理できるため、より高速に対応できる。この場合、1 秒で 100,000 カウント以上計測できていた。ただし、ロータリーエンコーダーをモーターと併用する場合、ノイズにより不要な割込みが発生することがあるため、速度よりも誤動作の少なさを重視し、モーター駆動のインバータ周波数を避けるなどの工夫が重要だと考える。

4. analogWrite

analogWrite は、新しいマイコンでは DAC (Digital to Analog Converter) が搭載され、任意の電圧を出力できる場合もある。しかしながら、Arduino における analogWrite は、疑似的なアナログ出力である PWM (Pulse Width Modulation) によって実現されているため、本稿ではこの PWM 出力について検討する。

Arduino の PWM 出力は、ピンによって約 490Hz または約 980Hz のいずれかの周波数に固定されており、ユーザーが制御できるのはデューティ比のみである。図 5 のようにコードを記述することで、PWM の設定そのものを変更できる。

周波数を高くする場合、タイマーの分解能が下がるため、デューティ比の設定可能な段階数が粗くなるが、高周波のパルス出力も可能である。これは、前述の digitalWrite と異なり、パルス出力中も他の処理を並行して行える点が魅力である。

```
void setup() {
  pinMode(9, OUTPUT);
  TCCR1A = B10000010;
  TCCR1B = B00011001;
  ICR1 = 15;
  OCR1A = 1;
}

void loop() {
}
```

図 5 1MHz のパルスを出力した際のコード

PWM 設定変更による高周波パルスの応用例として、筆者が試みた光通信によるシリアルデータ伝送について以下に述べる。

ノイズ対策の一環として図 6 に示すプラスチック光ファイバー^[2]によるシリアル通信を試みた。しかしながら、性能面では問題なかったが、コストや入手性に難があることが分かった。

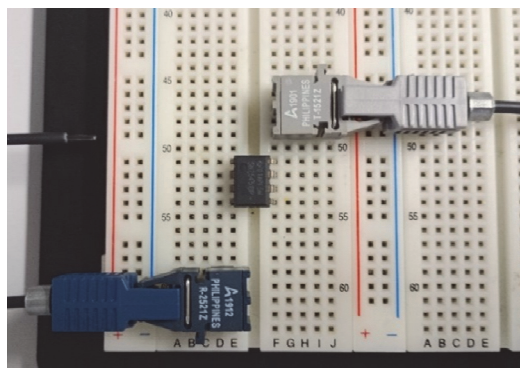


図 6 プラスチック光ファイバー

そこで、図 7 に示す、より入手しやすい光デジタルオーディオケーブル^{[3][4]}を用いたシリアル通信を検討した。ただし、このモジュールは PCM (Pulse Code Modulation) 信号などの音声用に設計されているため、信号が一度途切れるとモジュールの IC の動作が休止してしまい、断続的な通信には不向きであった。しかし、信号が連続していれば高周波のデータは通過したため、PWM

で生成した高速パルスとシリアル信号とをロジック IC で XOR 演算して送信したところ、通信に成功した。受信側には暫定的に RC ローパスフィルタを使用した。他により良い方法もありそうである。回路の模式図を図 8 に示す。

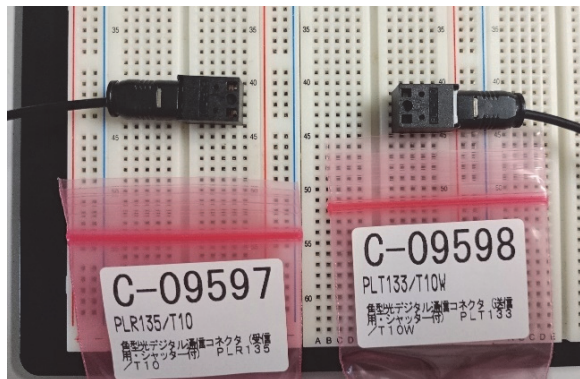


図 7 光デジタルオーディオケーブル

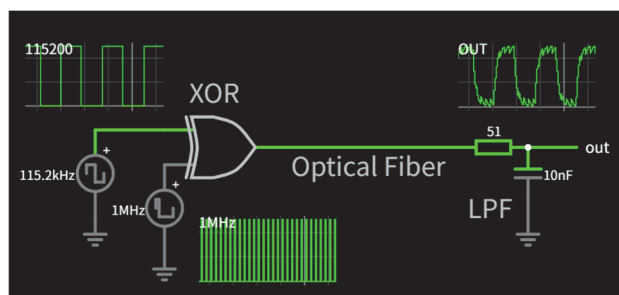


図 8 回路の模式図

5. analogRead

analogRead は、電圧を計測する機能であり、ポテンショメータによる電圧計測や、CdS セルでの照度測定、サーミスタによる温度計測などに使用される。処理時間は比較的長く、1 回あたり約 110 μ 秒 (約 8.9kHz) である。図 9 のコードでは、ADC (Analog to Digital Converter) にかかるクロックの分周比を 128 から 16 に変更しており、その結果、計測速度を 1 回あたり約 15 μ 秒 (約 66kHz) に高速化できた。これ以上の高速化も可能だが、明らかな測定誤差が生じるため、本稿ではこの程度の速度上昇が実用上適切であると考ええる。

この高速化により 20kHz 以上の周波数である超音波を利用した信号入力にも対応できる可能性があるため、超音波センサーを用いた測定システムを検討したことがある。しかし、実際に試作

したところ、ノイズの除去が難しく、また分解能も 10bit にとどまるため、微小信号の検出が困難であり、実用化には至らなかった。

```
void setup() {
  ADCSRA = ADCSRA & B11111000;
  ADCSRA = ADCSRA | B00000100;
}
// ADCSRAの下位3桁を111から100へ変更

void loop() {
  int value = analogRead(0);
}
```

図 9 高速化した analogRead のコード

6. おわりに

これらのコードは、Arduino 登場以前からマイコンを扱ってきた人にとっては当たり前の内容かもしれない。しかし、筆者の考えでは、Arduino は、初心者向けに複雑な部分を隠しつつ、使いこなすことでユーザー自身がマイコンの内部動作を理解し、より高度な活用へとステップアップできる仕組みがあると思われる。したがって、電子工作の学習において有用なツールであると考えている。

近年では、Arduino や他のワンボードマイコンの性能が飛躍的に向上している一方で、設計が複雑化している面もある。これからも電子工作の裾野が広がっていくことを願うとともに、初心者にもとつきやすく、より深く電子回路の世界に踏み込めるような、シンプルで魅力的な製品が登場することを期待したい。そして、マイコンで「何ができるか」を自分なりに発信し続けていければと思う。

参考文献

- [1] <https://docs.arduino.cc/resources/datasheet/s/A000066-datasheet.pdf>
- [2] <https://docs.rs-online.com/67fb/0900766b80ad66b3.pdf>
- [3] <https://akizukidenshi.com/goodsaffix/plr135-t10.pdf>
- [4] <https://akizukidenshi.com/goodsaffix/plt133-t10w-v3.pdf>