

§2-I-6 Arduino を用いた組み込みシステムの製作

(Embedded system with Arduino)

1. 実験の狙い

2-I-5 の実験において、様々なセンサから物理量を電圧値などで取得できることを確認した。本実験では、Arduino というデバイスを用いて、PC がセンサとデータのやり取りが行えることを確認する。

2. 解説

Arduino

Arduino は、2005 年イタリアの大学教授 Massimo Banzi 氏らによって、電気・電子の学生らが簡単に手に取って学べる安価なマイコンボード教材として開発されたものである。技術的なハードルがとて低いため、情報処理系や機械系、デザイナー系、クリエイター系などの学生や一般の人たちも利用するようになってきている。Arduino の大きな特徴は、オープンソースハードウェアの考えで開発されていることである。このため、Arduino の回路図や基盤図などが公開され、誰もが簡単に類似のクローン製品を開発することもできる。また、マイコンボードに組み込むためのソフトウェア開発環境（IDE：Integrated Development Environment 統合開発環境）も、無償でネット上からダウンロードして利用することができる。

Arduino は、実際のマイコンボードとその統合開発環境の 2 つを指すが、以降ハードウェアであるマイコンボードを Arduino、統合開発環境を IDE と略して解説する。Arduino には、Atmel 社のマイクロプロセッサが採用されており、Arduino Uno や Arduino Mega など多くの種類のものが製造・販売されている。本実験で使用する Arduino Uno は図 1 のような外観をしており、表 1 のような特徴を持つ。

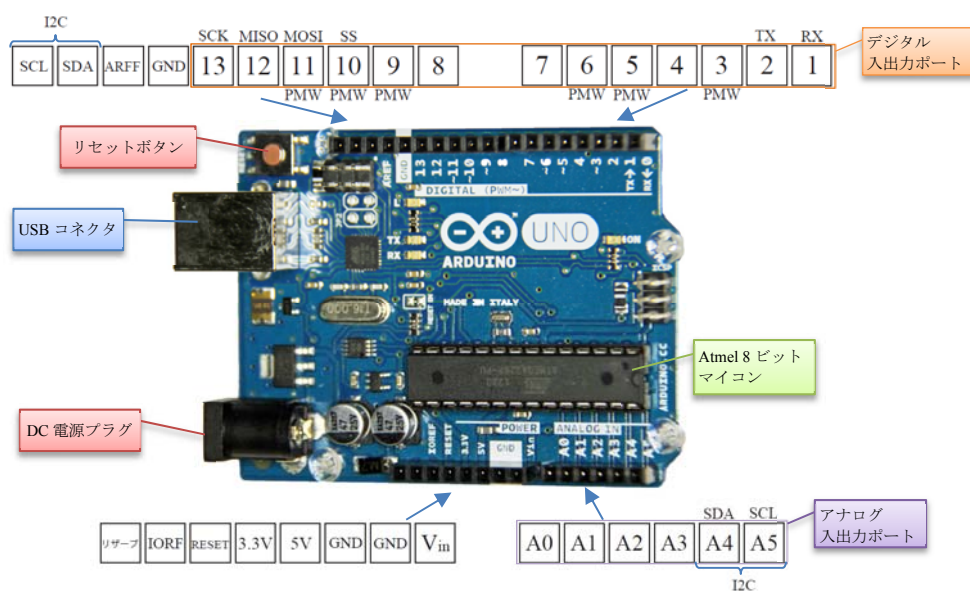


図 1 Arduino マイコンボードのインタフェース構成図

表 1 Arduino Uno の仕様

Arduino 仕様	値	Arduino 仕様	値
動作電圧	5 V	I/O ピン電流	40
推奨入力電圧	7～12 V	供給可能な最大電流	50 mA
制限入力電圧	6～20 V	フラッシュメモリ	32 kB
デジタル I/O ピン数	14	SRAM	2 kB
PWM チャネル	6	EEPROM	1 kB
アナログ I/O ピン	6	クロック周波数	16 MHz

Arduino のようなマイコンボードは、プログラムを組み込むことで、いろいろな動きや働きを与えることができる。Arduino には、外部の電子部品群の入出力機能を持ち、デジタルやアナログ入出力、シリアル通信処理ができる。また、プログラミング処理機能を持ち、計算処理や文字列処理などができ、割り込み処理機能なども持ち合わせている。さらに Arduino では、ピンの仕様やソフトウェアの仕様などが規格化されているため、Arduino 上に重ねて使う拡張ボード（シールド）も数多く製造・販売されている。

統合開発環境（IDE）

Arduino の統合開発環境は、図 2 のような画面のもので Java アプリケーションである。IDE を動かすために Windows 7 以上※の PC が必要となる。プログラミング言語は、C 言語に似ており書かれたプログラムはスケッチと呼ばれる。

※Windows 以外にも Linux や Mac OS なども利用可能である。

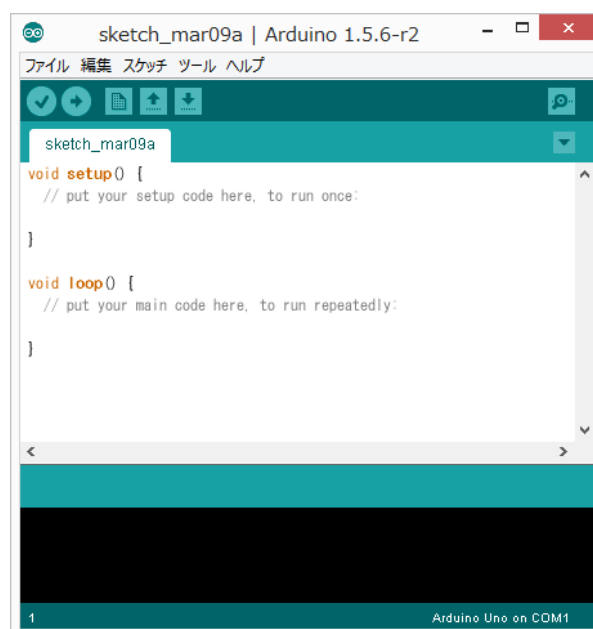


図 2 統合開発環境のソフトウェア画面

2. 実験の予習 *予習レポートとして整理して提出すること

予習1： 下記の用語について，意味を調査し理解を進めておく．

黒字は必ず調べること．その他は余裕があれば調べること．

- | | |
|-------------------|--------------------------------|
| a. Arduino | b. PWM |
| c. LCD | d. LED, フルカラーLED |
| e. Arduino IDE | f. processing (ソフトウェア) |
| g. ライブラリ | h. Funnel ライブラリ |
| i. delay 関数 | j. pinMode 関数, digitalWrite 関数 |
| k. analogWrite 関数 | l. analogRead 関数 |

予習2： Arduino Uno の基板構成（どのようなピンがあるかなど）を必ず調べ来ること．

※役割や機能など．理解をしていないと，実験がスムーズに行えません。

予習3： LED マトリックスの制御では，基本的なプログラムを基に LED 点灯パターンに合わせたプログラムを作成してくること．

予習4： C 言語をよく復習してくること．

※予習1のi～jはArduino IDEで使用する関数である．どのような機能があるのか，予習2のどのピンを使用のかなど調査すること．

3. 実験

3.1 実験の概要

Arduino IDE を用いて，Arduino にプログラムを書き込み，様々な素子を制御する．

実験は例題に沿って行う．

- LED の点灯制御
- PWM による輝度の調整
- デジタル入力による LED の点灯制御
- PWM の輝度値計測
- LED マトリックスの制御（できるところまで）

実験の詳細に関しては Moodle にアップロードしてある別資料の手順を確認しながら行う．

3.2 実験で使用する機材

- ・ Arduino UNO
- ・ パーソナルコンピュータ
- ・ センサ

実験では主に図 3 に示すような測定系を用いる．

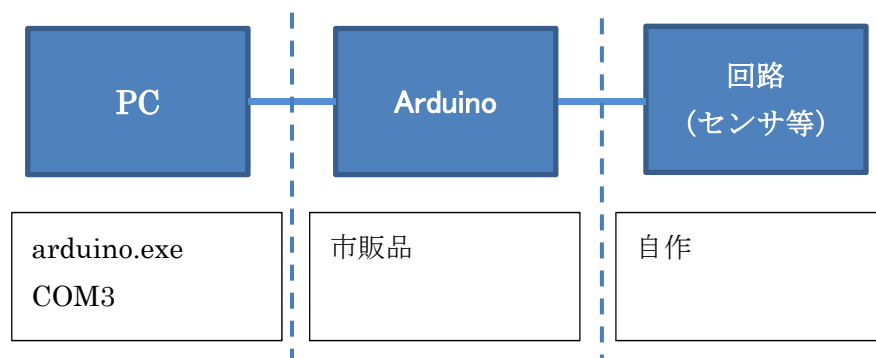


図 3 実験の測定系

実験室の PC には（各自が持参した）USB を接続しないこと

3.3 実験手順

[1] 実験の進め方

～13:45	手順 1：Arduino IDE の設定 [資料 pp. 1-2] 手順 2：例題 1 の実施
～14:20	手順 3：例題 2 の実施 [資料 p. 3]
～14:40	手順 4：例題 3 の実施 [資料 p. 4]
～15:00	手順 5：例題 4 の実施 [資料 p. 5] 各センサに関してブレッドボードを用いて回路を作成する。
～15:30	手順 6：例題 5 の実施 [資料 p. 6]
～16:10	手順 7：例題 6 の実施 [資料 p. 7-8] LED の点灯パターンの制御はプログラムかんがえてくること。
～16:20	手順 8：討論
～16:25	手順 9：TA による試問

[2] 各プログラムの詳細

- 各プログラムの詳細は Moodle 上に上がっているので、実験前に各自が確認をしておくこと。

4. 実験レポート

検討すべき事項（考察事項）

実験レポートでは、実験結果を踏まえた検討を行い、下記の項目に関してもよく考察をすること。

- ① PWM とアナログ制御の違いについて。
- ② LED マトリックスはどのように制御することが望ましいか。
- ③ センサを利用したシステムとしてどのようなものが考えられるか。
システム構成などを考えてみよう。

課題

課題1： Arduino Uno のスペックを調査せよ。

このデバイスを利用する際に注意しなければならないことを述べよ。

例としてどのようなシステム(プログラム)を考えるとときにそのような問題が発生するか。

課題2： シリアル通信の特徴について調べよ。

課題3： LED マトリックスにおいて、制御する LED を増やしていくとどのような問題が発生するか考えよ。

(LED を積層させた LED CUBE (8 x 8 x 8) 作る場合 Arduino と LED を接続するだけで問題ないか?)

5. 参考文献

- ・ 小林茂 著「Prototyping Lab」(オーム社)
- ・ 船田巧 著「Arduino をはじめよう」(オーム社)

1. 準備

- 1) デスクトップ上の「arduino.exe」を起動する．(プログラム名：Arduino IDE)
- 2) 図 4 のような画面が起動されることを確認する．



図 4 Arduino IDE の起動画面

- 3) ポートの確認
コンピュータ(右クリック)>プロパティ>デバイスマネージャ を選択
ポート (COM と LPT) を選択し，Arduino Uno のポート番号を確認する．
※本書では「COM3」として以下進める．
- 4) Arduino IDE のポート設定
ツール>シリアルポート>COM3 を選択．
また，ツール>Board>Arduino Uno を選択する．
※上記は既にインストールしてあるものを利用するための設定である．個人で Arduino を使用したい場合は，書籍や Web サイトを参考にプログラムやデバイスドライバのインストールなど設定を行うこと．
- 5) 初期画面上のブロックの説明
setup：動作開始時に一度だけ実行される
loop：Arduino ボードの電源がオンである限り，繰り返し実行される
- 6) 基板上の LED の制御
図 5 のプログラムを記入しマイコンボードに書き込み（図 4 の書き込みボタンを押す）基板上にある LED が点滅することを確認する．

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(500);
  digitalWrite(ledPin, LOW);
  delay(500);
}
```

図 5 ボード上の LED 制御用コード

2. プログラムの作成

[1] 例題 1：LED の点灯制御

- 1) 図 6 のような回路を作成する．（抵抗は 330Ω ）

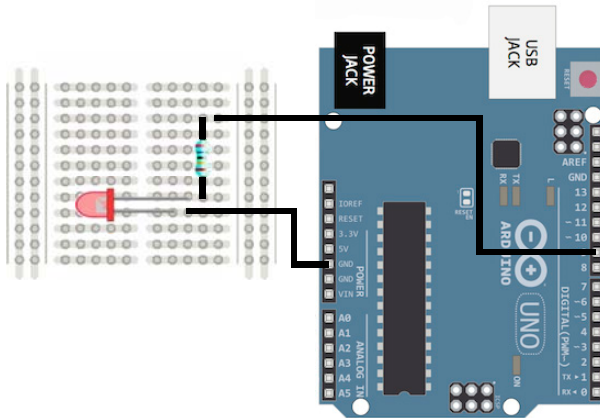


図 6 LED 点灯制御用回路

- 2) 図 5 の “ledPin = 13” を “ledPin = 9” に変更してマイコンボードに書き込む．LED が点滅することを確認する．
- 3) 図 7 の回路を作成する．（抵抗は 330Ω ，カソードコモン LED を使用）

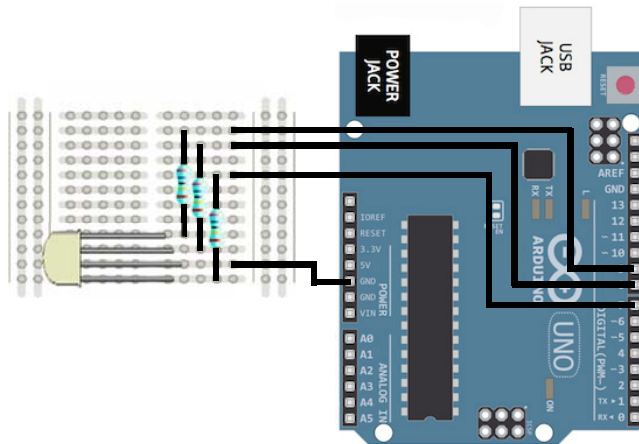


図 7 フルカラーLED 点灯制御用回路

- 4) 図 8 のプログラムを記入しマイコンボードに書き込み

```
int RledPin = 9;
int BledPin = 8;
int GledPin = 7;

void setup() {
  pinMode(RledPin, OUTPUT);
  pinMode(GledPin, OUTPUT);
  pinMode(BledPin, OUTPUT);
}

void loop() {
  digitalWrite(RledPin, HIGH);
  digitalWrite(GledPin, HIGH);
  digitalWrite(BledPin, HIGH);
}
```

図 8 フルカラーLED 制御コード

- 5) Loop 内のそれぞれの Pin の HIGH と LOW を入れ替えて LED の光り方が変わることを確認せよ．

[2] 例題 2：PWM による輝度の調整

1) delay による輝度調整

図 6 の回路に作り直す。

この時作成したプログラム(図 5)を開きなおす。

delay のカッコ内の数字 (500 と 500) を (5 と 5) および (1 と 9) に変更して、明るさの違いを観察する。(照度計を用いること.)

2) アナログ出力による制御

D9 端子に 0～255 (5 ステップ) のアナログ出力を行うため、図 9 のプログラムをマイコンボードに書き込む。

```
int ledPin = 9;

void setup() {
}

void loop() {
  for(int value = 0; value <= 255; value += 5){
    analogWrite(ledPin, value);
    delay(30);
  }
  for(int value = 255; value >= 0; value -= 5){
    analogWrite(ledPin, value);
    delay(30);
  }
}
```

図 9 アナログ出力による輝度調整

※アナログ出力が可能なピンは PWM の記載のある D3, D5, D6, D9, D10, D11 である。

[3] 例題 3： デジタル入力

- 1) 図 10 の回路を作成する（抵抗は 330Ω , $10\text{ k}\Omega$ ）

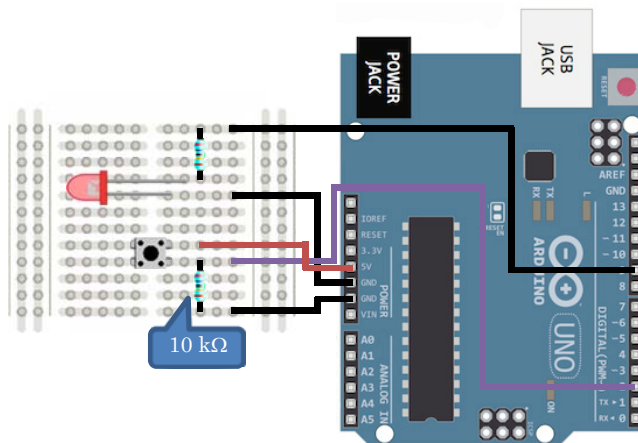


図 10 デジタル入力確認用回路

- 2) アナログ出力による制御

図 11 のプログラムをマイコンボードに書き込む。

```
const int buttonPin = 2;
const int ledPin = 9;
int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if(buttonState == HIGH){
    digitalWrite(ledPin, HIGH);
  }
  else{
    digitalWrite(ledPin, LOW);
  }
}
```

図 11 デジタル入力による LED 点灯

- 3) 基盤のスイッチを押すと LED が点灯することを確認する。

ここまでの実験で、Arduino を用いたデジタルの入出力、アナログ（PWM）の出力の手法を学んだ。Arduino にはこの他にもアナログ入力の端子があり、アナログ計測などに用いることが可能である。次項でその例を説明する。

[4] 例題 4：アナログ入力によるデータ取得とシリアル通信

- 1) 図 12 のような回路を作成する。

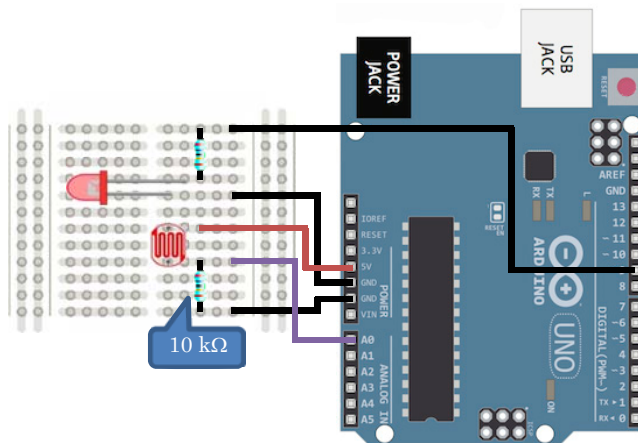


図 12 アナログ入力確認用回路

- 2) 図 13 のプログラムをマイコンボードに書き込む。

```
const int sensorPin = 0;
const int ledPin = 9;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int brightness = analogRead(sensorPin);
  int intensity = map(brightness, 0, 1023, 255, 0);
  analogWrite(ledPin, intensity);
  delay(50);
}
```

図 13 CdS センサーによる輝度の取得

- 3) 上記のプログラムにより、輝度に応じたアナログ入力値が得られる。明るさが変わると LED の輝度が変わることを確認することができる。
- 4) 図 13 のプログラムを図 14 のプログラムに変更し、マイコンボードに書き込む。

```
const int sensorPin = 0;
const int ledPin = 9;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int brightness = analogRead(sensorPin);
  Serial.println(brightness);
  int intensity = map(brightness, 0, 1023, 255, 0);
  Serial.println(intensity);
  analogWrite(ledPin, intensity);

  delay(50);
}
```

図 14 シリアル通信の例

- 5) プログラムが書き込まれたら、メニューバーの ツール>シリアルモニター を選択する。
“Serial.println”で指定された値が表示されることが確認できる。

[5] 例題 5: PWM の輝度値計測

- 1) 本実験では例題 2 と例題 4 の実験を組み合わせて、PWM により輝度を変化させた時の CdS セルの値の変化をグラフ化する。
- 2) プログラムに関しては以下の測定項目に応じて書き換えること。
- 3) CdS セルの測定値を計測する。
 - i. “analogWrite”の値を 0～255 までを 5 刻みで変化させ、CdS セルから得られる 0～1023 の範囲の値を、シリアル通信を利用して取得する。
 - ii. “digitalWrite”での出力（HIGH と LOW）を delay 関数の比率を変化させて、i. 同様シリアル通信を用いて CdS セルからの値を取得する。
- 4) ※1 測定は外光が入りにくいよう、箱の中で行うこと。
※2 測定終了後グラフを作成すること。

[6] 例題 6: LED マトリックスの制御（実験時間超過の場合は行わない）

- 1) 本実験では図 15 のような回路を使用する。

図中の赤線がアノード，黒線がカソードである。

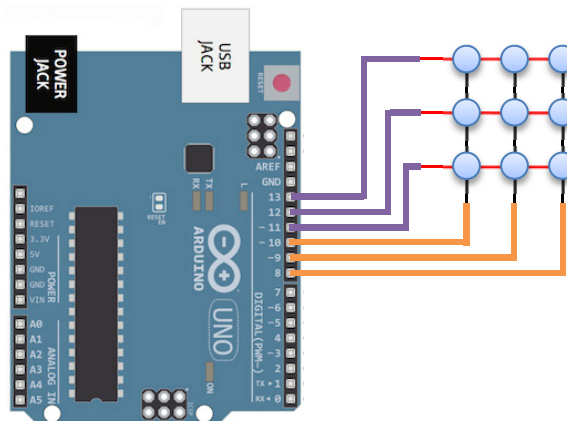


図 15 LED マトリックス

- 2) LED を光らせる

下図のようなプログラムをマイコンボードに書き込む。

LED が 1 つと 1 列それぞれ点灯することを確認する。

```
const int ledPin13 = 13;
const int ledPin12 = 12;
const int ledPin11 = 11;
const int ledPin10 = 10;
const int ledPin9 = 9;
const int ledPin8 = 8;

void setup() {
  pinMode(ledPin13, OUTPUT);
  pinMode(ledPin12, OUTPUT);
  pinMode(ledPin11, OUTPUT);
  pinMode(ledPin10, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(ledPin8, OUTPUT);
}

void loop() {
  digitalWrite(ledPin13, LOW);
  digitalWrite(ledPin12, LOW);
  digitalWrite(ledPin11, HIGH);
  digitalWrite(ledPin10, HIGH);
  digitalWrite(ledPin9, HIGH);
  digitalWrite(ledPin8, LOW);
  delay(100);
}
```

図 16 LED を 1 つ光らせるプログラム

```
const int ledPin13 = 13;
const int ledPin12 = 12;
const int ledPin11 = 11;
const int ledPin10 = 10;
const int ledPin9 = 9;
const int ledPin8 = 8;

void setup() {
  pinMode(ledPin13, OUTPUT);
  pinMode(ledPin12, OUTPUT);
  pinMode(ledPin11, OUTPUT);
  pinMode(ledPin10, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(ledPin8, OUTPUT);
}

void loop() {
  digitalWrite(ledPin13, LOW);
  digitalWrite(ledPin12, LOW);
  digitalWrite(ledPin11, HIGH);
  digitalWrite(ledPin10, LOW);
  digitalWrite(ledPin9, LOW);
  digitalWrite(ledPin8, LOW);
  delay(100);
}
```

図 17 LED を 1 列光らせるプログラム

- 3) LED を光らせるパターンを考える。

下図に示すパターンが順に光るプログラムを考えよ。

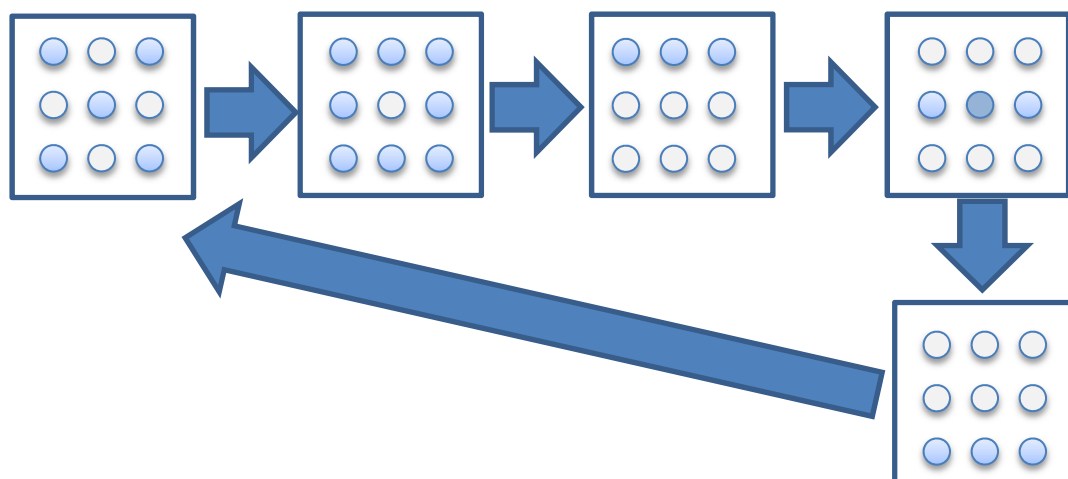


図 18 LED 点灯パターン