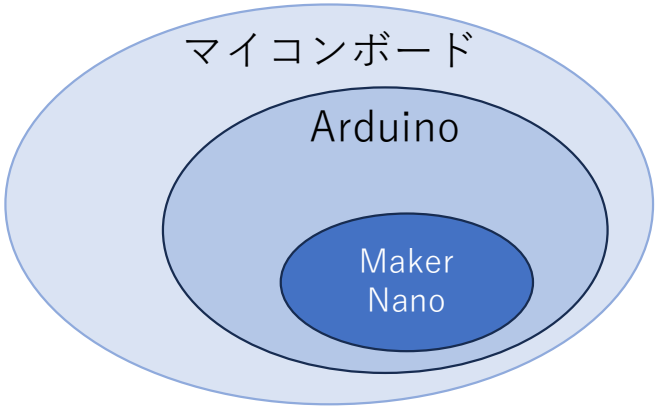
プログラミング実習(1)

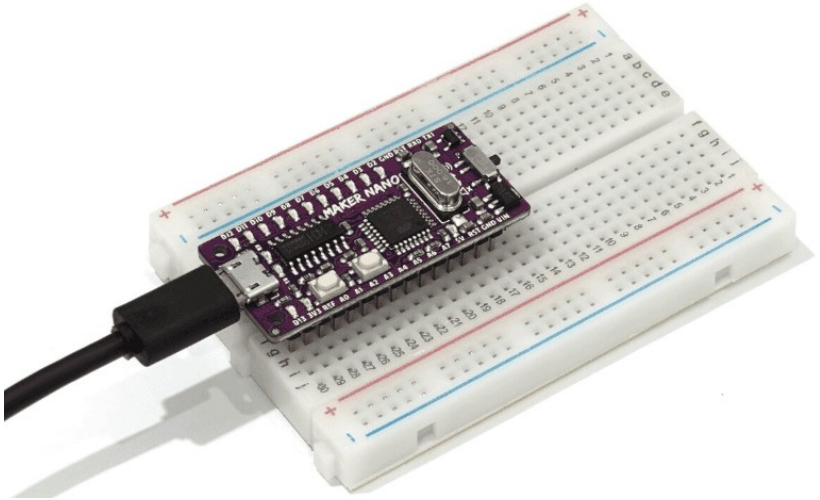
マイコンを使ってプログラミングの基礎を学ぶ

① Arduino プログラミングの基礎

マイコンボードの接続

- この授業では、「Maker Nano」という「マイコンボード」を使います。
- 以下、単に「マイコンボード」と呼ぶことにします。
- プログラミングの文法などの説明や、Arduinoシリーズおよび互換機に共通する事柄の説明において「Arduino」と呼ぶこともあります。





ただし、ボードに搭載されているLEDやスピーカーなど Maker Nano固有の事情を説明する場合には「Maker Nano」と標記します。

プログラミング開発環境について

- Arduino IDEのインストール (事前準備)
- ボードの設定、下記の通り選ぶ。
  - 「ツール」-「ボード」-「Arduino AVR Boards」- 「Arduino Nano」
    ※ Maker Nanoは、 Arduino Nano互換機

- USBポートの設定
  - マイコンボードを繋がない状態で
  - 「ツール」-「ポート」を開き、このときに現れるCOMポートを確認する
  - マイコンボードをUSBポートに繋ぐ。
  - Windows の場合、マイコンを繋ぐと 「COM3」 (PCにより異なる)などと新たなポートが現れる。マイコンを繋いだ場合に現れたポートがマイコンボードとの通信で使うポートである。

- Macを使う場合の補足(ドライバのインストールの必要性)

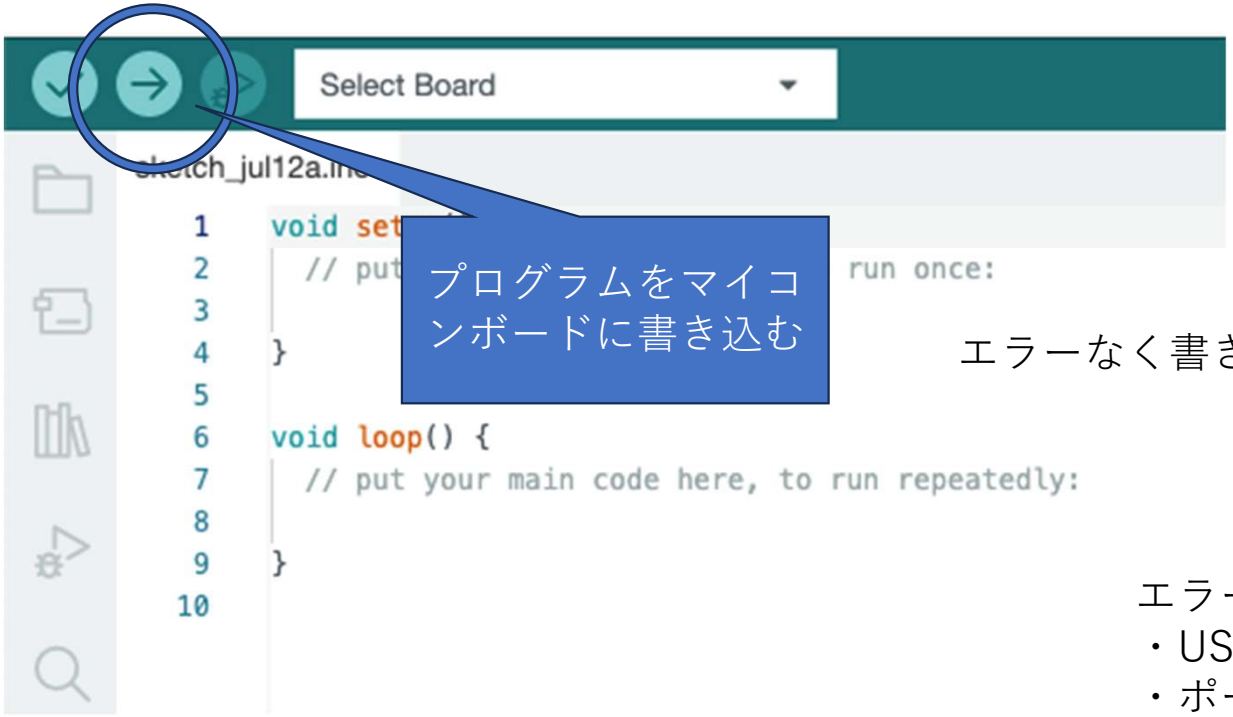
Arduino開発環境の確認

IDEを開いたとき、次のようなウィンドウは開いただろうか。開いていなければ、
「ファイル」-「新規」を選んで、表示させてください。

Arduinoのプログラムのことを「スケッチ」と呼びます。上の図にあるものが
「スケッチ」の骨組みです。

Arduino開発環境の確認

このスケッチは、処理する内容がまったく記述されていない空っぽのスケッチです。
これを書き込んで、プログラムの書き込みがうまくいくかどうか確認してください。



エラーなく書き込みが完了すればOK

エラーが起きた場合は以下を確認
・USBケーブルが挿入されているか
・ポートが正しく選択されているか

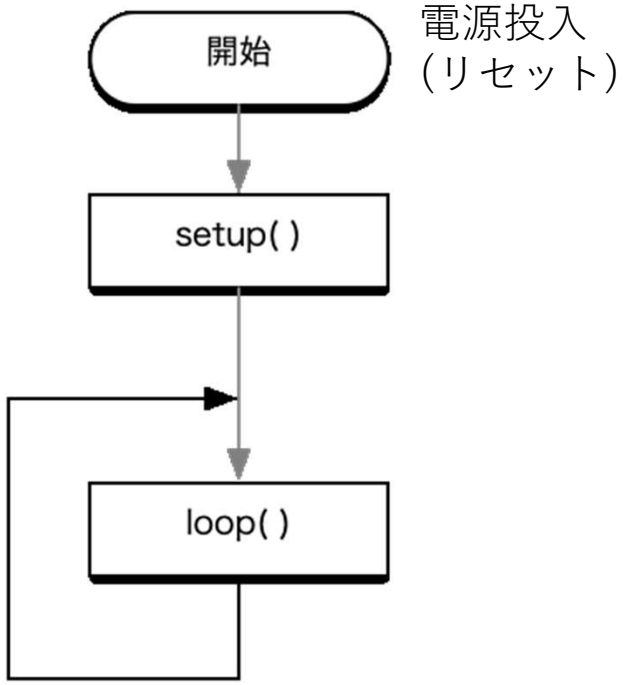
スケッチの骨組み

```
void setup(){
    マイコンが起動したとき、最初に1回行う
    処理をここに記述します。
    通常、マイコンに接続するセンサーやデバイスの
    初期化を行います。
}

void loop(){
    ここに記述された内容は電源が切れるまで、
    または、リセットが押されるまで繰り返し実行されます。
}
```

処理内容を記述

マイコンボードの動作



プログラムの構成を確認する(1)

setupの部分が一度だけしか実行されないことを確認します。
プログラムで使われている数字などの意味は後で説明しますが、440は周波数で、楽器のチューニングで使われる「ラ」の音です。これを短く発します。

prog1.ino

```
void setup() {
      tone(8, 440, 50);
}

void loop() {
}
```

左のように入力できたら
prog1 という名前で保存してください。

ファイルは prog1.ino という名前になっています。

もう一度、音を確認するには、マイコン上のリセットボタンを押します。

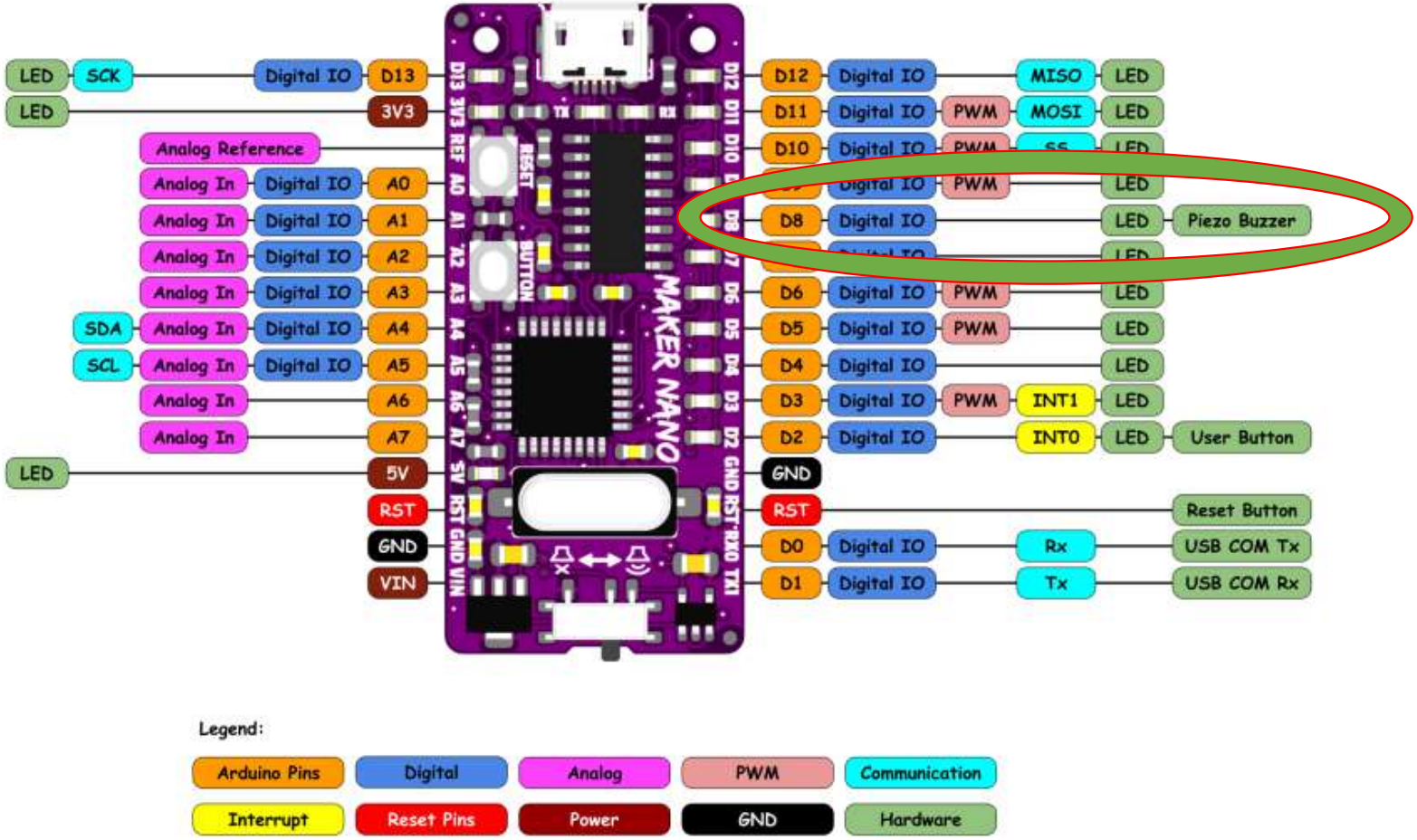
実行時の様子について

実行すると、音がなるのと同時に、一つのLEDが点滅します。
よく見ると、D8でという名前が付いています。

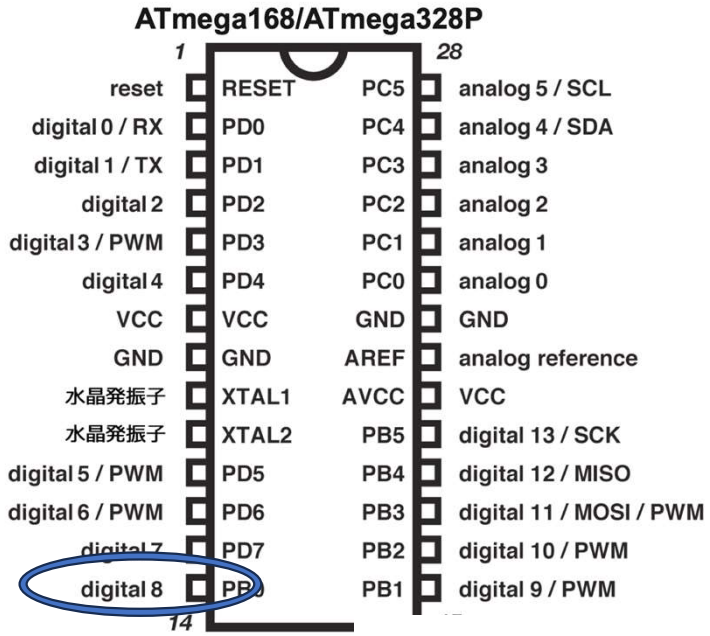
```
tone(8, 440, 50);
```

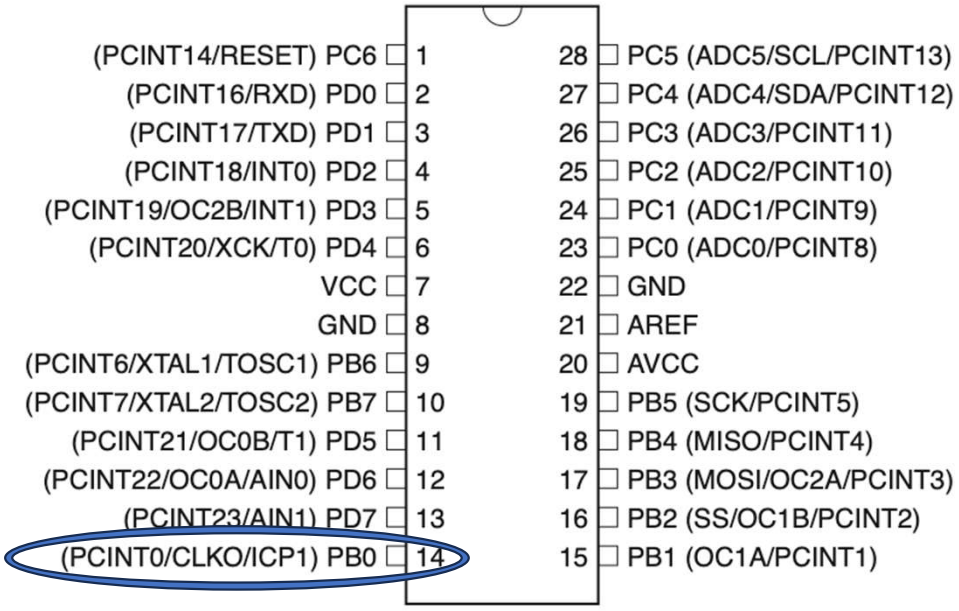
ピン8 を通して 440Hzの音を 50m秒の長さで音をならす。
ピン8への出力により、ここに繋がっているLEDも一瞬点灯させます。

ピン8

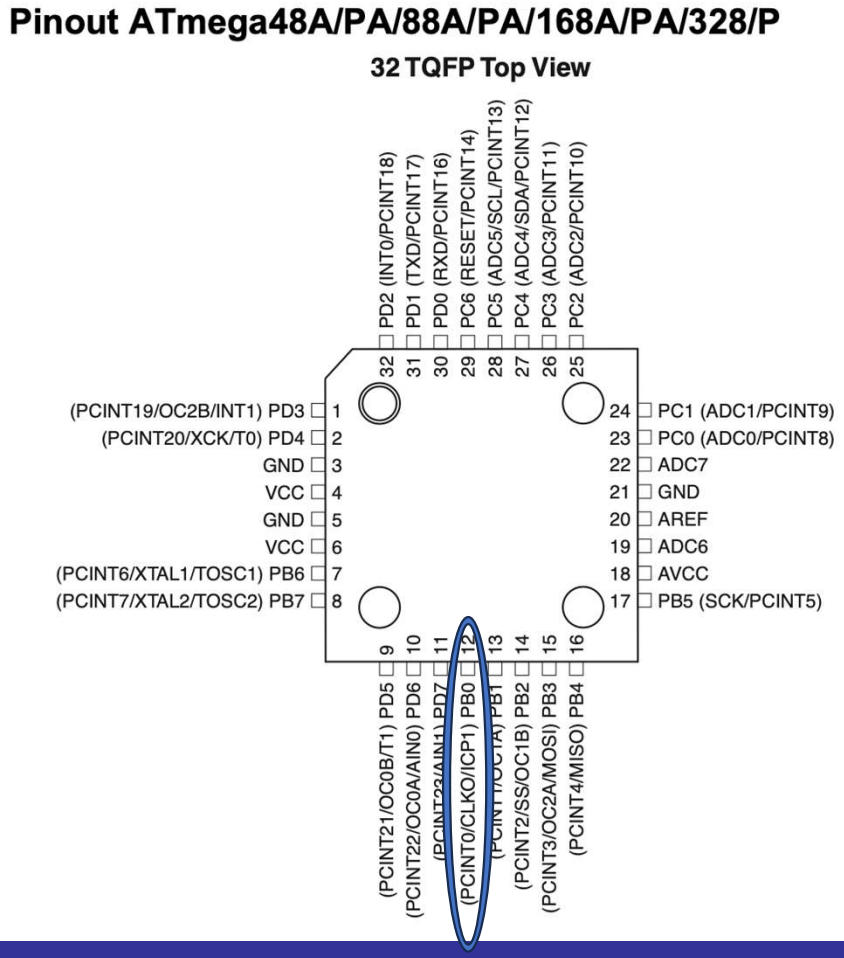

LED
SCK
Digital IO
D13
LED
3V3
Analog Reference
Analog In
Digital IO
A0
Analog In
Digital IO
A1
Analog In
Digital IO
A2
Analog In
Digital IO
A3
SDA
Analog In
Digital IO
A4
SCL
Analog In
Digital IO
A5
Analog In
A6
Analog In
A7
LED
5V
RST
GND
VIN
D12
Digital IO
MISO
LED
D11
Digital IO
PWM
MOSI
LED
D10
Digital IO
PWM
LED
D8
Digital IO
LED
Piezo Buzzer
D6
Digital IO
PWM
LED
D5
Digital IO
PWM
LED
D4
Digital IO
LED
D3
Digital IO
PWM
INT1
LED
D2
Digital IO
INT0
LED
User Button
GND
RST
Reset Button
D0
Digital IO
Rx
USB COM Tx
D1
Digital IO
Tx
USB COM Rx
MAKER NANO
RESET
BUTTON
Legend:
Arduino Pins
Digital
Analog
PWM
Communication
Interrupt
Reset Pins
Power
GND
Hardware

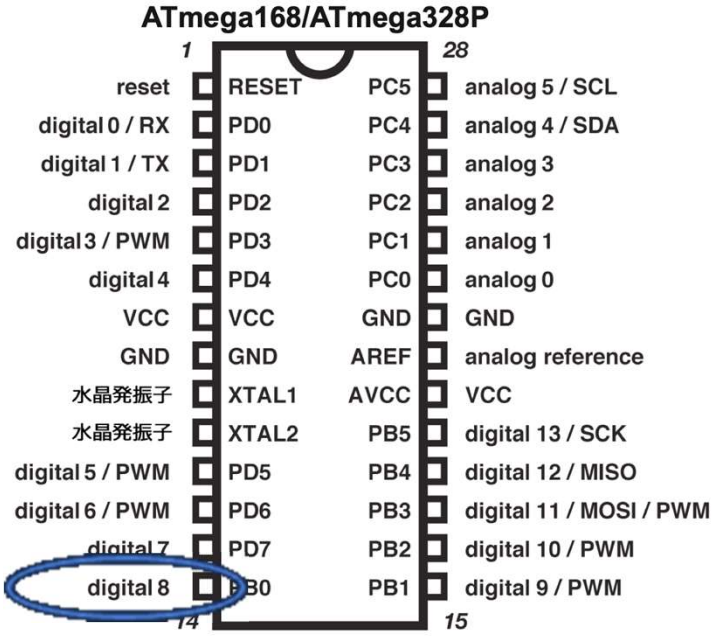
プログラムで使うピンとチップのピン





プログラムのピンと回路





図のマイコンのピンが マイコンボードのD8のところにつながっていて、プログラムの中でこのピンに信号を送ったり、信号を受けとったりできます。

プログラムの構成を確認する(2)

loop()にコードを書き加える効果についても確認しておこう

prog1.ino（修正-step 1）

```
void setup() {
      tone(8, 440, 50);
      delay(1000);
}
void loop() {
      tone(8, 262, 50);
      delay(1000);
}
```

delay(長さ)

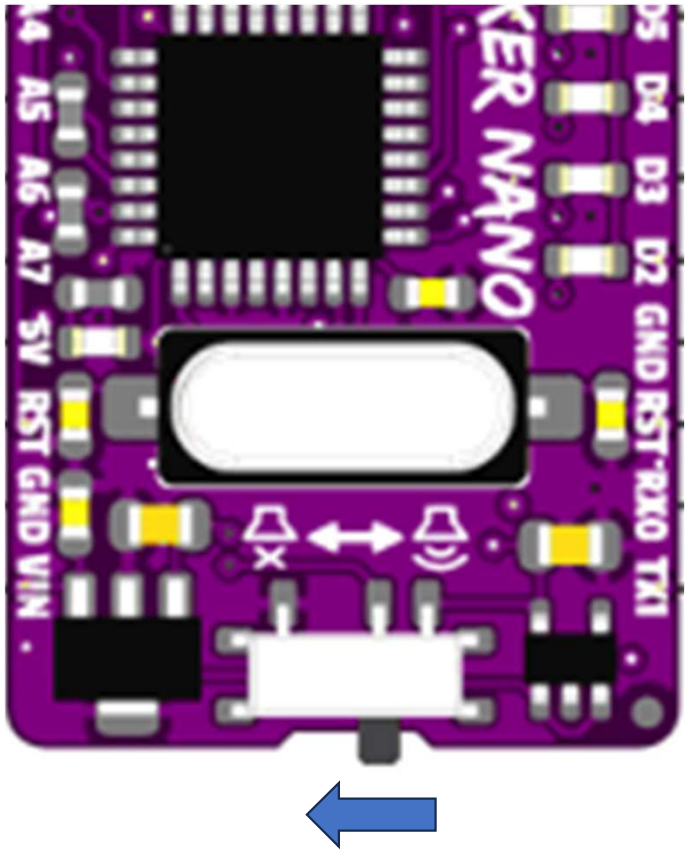
次の命令に進むまでの待ち時間をミリ秒単位で指定します。

262Hz は 「ド」の音です。

マイコンにプログラムを書き込み、実行する前に、どんな振る舞いをするか想像してみましょう。

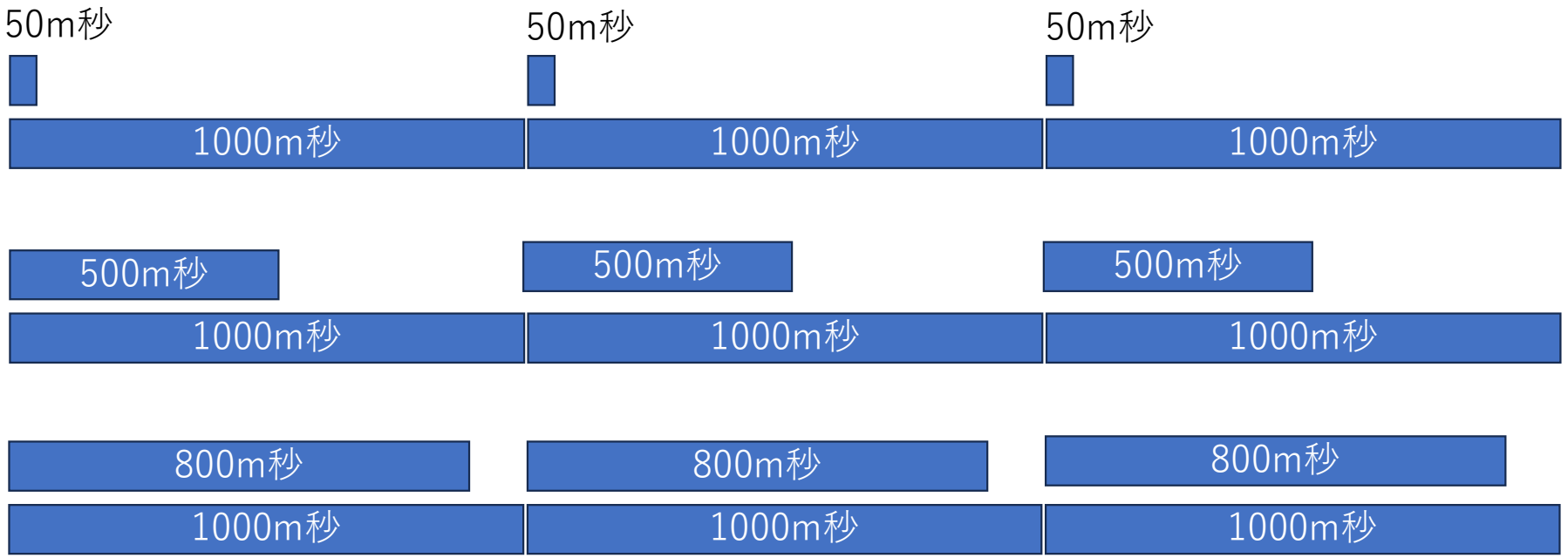
スピーカーのスイッチ

- 私たちの使っているマイコンにはスピーカーのオン/オフのスイッチがあります。
- 動作が分かったらスピーカーをオフにしておきましょう。
- また、「新規スケッチ」を作成し、空のプログラムを転送してマイコンの動作を止めておく(?)のもよいでしょう。



(補足)音とタイミング

※ toneは、音を鳴らし始めたら、すぐ次の命令に移ります。toneの音が鳴り終わってから次の行の命令に進むわけではありません。(toneは非同期)



先ほど作ったプログラムを書き換えて上記のことを確かめてみてください。

プログラムの基本

プログラムの基本を学ぶために、当面、一度だけ実行される setup()の方だけ使うことにします。

次のプログラムを参考に 「ド」「レ」「ミ」を順にならしてください。

prog2.ino

```
void setup() {
      tone(8, 262, 480);
      delay(500);
      加筆してください
}
void loop() {}
```

テンポを 1分間に120とする
60秒で120回 なので、 1回あたり0.5秒
delayでタイミングを計ると500m秒ずつ待ち時間をとると良さそう。

音階	周波数
ド	262
レ	294
ミ	330
ファ	349
ソ	392
ラ	440
シ	494
ド	523

delay(500) は 500m秒、次の処理に進むのを待つ関数です。

プログラムの基本

prog2.ino

```
void setup() {
      tone(8, 262, 480);
      delay(500);
      tone(8, 294, 480);
      delay(500);
      tone(8, 330, 480);
      delay(500); //余分
}
void loop() {}
```

テンポを 1分間に120とする
60秒で120回 なので、 1回あたり0.5秒
delayでタイミングを計ると500m秒ずつ
待ち時間をとると良さそう。

音階	周波数
ド	262
レ	294
ミ	330
ファ	349
ソ	392
ラ	440
シ	494
ド	523

関数の定義

- 前のプログラムのようにして、もっと長いメロディをならそうとすると、そのたびに、toneとdelayをならさなければならない。
- なんども繰り返すような処理は、自分で「関数」として定義して、簡単に呼び出せるようにすることができる。

関数の定義を用いた例

```
void setup() {
      myTone(262);
      myTone(294);
      myTone(330);
}
void loop(){}

void myTone(int freq){
    tone(8, freq, 480);
    delay(500);
}
```

Arduinoのプログラミング言語は
C や C++というポピュラーな言語から
派生したものです。

処理内容は、基本的に、動作を指示する
関数を列挙することで記述します。

関数の定義の詳細は後に行います。

ここでは、私たち利用者も、マイコン
ボードにやらせたい一連の処理に
名前をつけておき、簡単に呼び出せる
ようにできるということだけ、みておいて
ほしい。

関数の定義を用いた例

```
void setup() {
      myTone(262); myTone(294);
      myTone(330);
}
void loop(){}

void myTone(int freq){
     tone(8, freq, 480);
     delay(500);
}
```

先に進む前に

プログラムの記述法の注意

関数を定義の仕方と呼び出し方の確認

関数名、変数名の注意

変数とデータ型

デジタルIO

デジタルIO

- tone()関数を使うときには、省略しましたが、通常、マイコンボードのデジタル入出力ピン(D0～D13)から信号を出し入れするには、次のような手順を行います。

- setup()の定義の中で、pinMode()関数を使い、入力・出力のいずれで使うかを指定する。

- pinModeを使った後で、 以下の関数を用いる。
  出力には digitalWrite()
  入力には digitalRead()

- これらの詳しい文法的な説明は、あとで行う。
- 取り扱われる信号はデジタルなので HIGH(5V) / LOW(0V)のいずれかになる。

デジタル入出力の設定

- Maker Nanoには、デジタル入出力に使えるピンがD0～D13の14個あります。

番号	用途
0	シリアル通信RXD
1	シリアル通信TXD
2	ボタンに接続
3	PWM*
4	
5	PWM
6	PWM

番号	用途
7	
8	ブザーに接続
9	PWM
10	PWM
11	PWM
12	
13	(Arduino互換機の多くはLEDに接続)

- PWM (Pulse Width Modulation)の注記のあるピンは、擬似的なアナログ入出力が可能で0-255の段階的な値を取り扱うことができます。
- その他はデジタル入出力専用で 0 (LOW=0V)か 1(HIGH=5V)のどちらかの値となります。

pinMode()の使用例

blink_sample.ino

```
void setup(){
     pinMode(13, OUTPUT);
}

void loop(){
     digitalWrite(13, HIGH);
     digitalWrite(13, LOW);
}
```

このプログラムは、loop()の中で
ピン13へ、HIGHとLOWの出力を
繰り返しています。

D13のLEDがどのようになるか注目してみてください。

※ 残念ながら、この例では ON/OFFの切り替えが早すぎて人間の目には点滅しているように見えません。
次のスライドの改良で点滅の様子がわかるように修正します。

pinMode()の使用例

blink_sample.ino

```
void setup(){
     pinMode(13, OUTPUT);
}

void loop(){
     digitalWrite(13, HIGH);
     delay(500);
     digitalWrite(13, LOW);
     delay(500);
}
```

このプログラムは、loop()の中で
ピン13へ、HIGHとLOWの出力を
500m秒の間隔をおいて、繰り返しています。

D13のLEDがどのようになるか注目してみてください。

pinMode()の使用例 – 処理する順番に記述

blink_sample.ino

```
void setup(){
    pinMode(13, OUTPUT);
}

void loop(){
    digitalWrite(13, HIGH);
    delay(500);
    digitalWrite(13, LOW);
    delay(500);
}
```

プログラムを記述するときには、
マイコンに処理を行わせたい順番に
処理内容を並べます。

順番が大事です。

digitalWrite(13, HIGH);
digitalWrite(13, LOW);
delay(500);
delay(500);

では、意味がありません。

でも、
digitalWrite(13,LOW);
delay(500);
digitalWrite(13,HIGH);
delay(500);
なら、まあまあ良いかもしれません。

出来たら、D12 など、他の LEDが点滅するようにプログラムを修正してみなさい。

デジタル入出力の設定(2)

文法

pinMode(pin, mode)

パラメータ

pin ： 設定を行うArduinoのピン番号

mode： INPUT, OUTPUT, INPUT_PULLUP のいずれか

Arduinoのピンは、何も指定しなければ入力用として使える。

https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/

デジタル出力 digitalWrite()

指定したデジタルピンに、HIGHかLOWを出力する。

文法

digitalWrite(pin, value)

パラメータ

pin ： 設定を行うArduinoのピン番号

value：HIGH （1=5V(マイコンにより3.3V)）
　　LOW　（0=0V）

https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/