



— 目次 —

ArduBlock とは .....	1
「ArduBlock」をダウンロードする。 .....	1
「ArduBlock」と「Arduino IDE」 .....	2
第1章 コンピュータ制御の基礎 .....	3
制御 .....	3
計測・制御システム .....	4
制御の種類 .....	6
アルゴリズムとフローチャート .....	7
プログラムとプログラム言語 .....	9
マイコンとプログラム .....	10
第2章 時計付き電子オルゴールの製作 .....	13
第3章 プログラムによる制御 .....	18
制御学習の準備をしよう .....	18
アルドゥブロックの基礎 .....	19
主なブロックのはたらき .....	20

## 1. ArduBlock(アルドゥブロック)とは

この学習で使うフリーソフトです。無料でダウンロードできますので、各家庭のパソコンでも、学校と同じように学習できます。

## 2. 「ArduBlock」をダウンロードする。

①ヒダピオの専用WEBページを開きます。

「ヒダピオ」で検索してみましょう！

②メニュー内の「ArduBlock」をクリックします。

③ファイルのダウンロードサイトをあげ、(arduino-\*\*\*-ArduBlock-\*\*\*.zip)をダウンロードし、任意のフォルダに保存します。

arduino-\*\*\*-ArduBlock-\*\*\*.zip … \*\*\*はバージョンによって異なります。

(任意のフォルダの例：マイドキュメントやUSBメモリ)

保存したファイル(arduino-\*\*\*-ArduBlock-\*\*\*.zip)を解凍してください。

プログラムを組もう .....	22
センサを使おう .....	25
関数（サブルーチン）を使おう .....	26
第4章 いろいろな制御 .....	29
デジタル数字 .....	29
信号機 .....	31
N ゲージ .....	33
電子オルゴール .....	34
参考資料 .....	35
2 進数 .....	35
2 進数とコンピュータ .....	36
表計算ソフトでのプログラム例 .....	36
HiDaspx .....	36
ATtiny2313 とマイコンボード .....	37
謝辞 .....	37

### 3. 「ArduBlock」と「Arduino IDE」

Arduino は、「初心者でも簡単に扱えるマイコンボード（AVR マイコン、入出力ポートを備えた基板）」＋「C++風の Arduino 言語」と、それらの統合開発環境から構成されるシステムのことです。また ArduBlock は Arduino（アルドゥイーノ）の初心者向けに開発された視覚的プログラム言語で Arduino IDE のツールの一つです。

ArduBlock で「アップロード」ボタンを押すと、Arduino でスケッチと呼ばれるプログラムにコンパイル（翻訳）され、さらに Arduino IDE 機械語にコンパイルされマイコンに書き込まれます。

ここで使用する ArduBlock は、HiDaspx（ヒダピオ学習回路）を使った中学校の「プログラムと計測制御」学習がスムーズに行えるように改変されています。

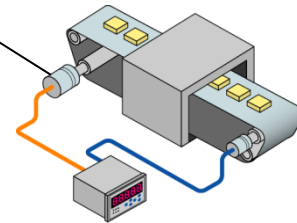
# 第1章 コンピュータ制御の基礎

## 1. 制御

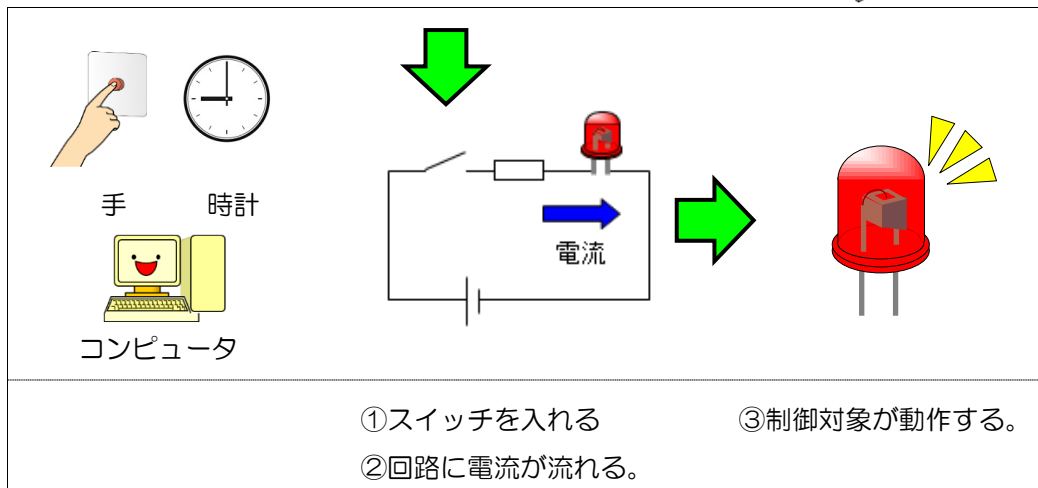
### 制御対象と制御

発光ダイオードやモータなどの制御する機器を( **制御対象** )といい、スイッチを( **ON-OFF** )するなどして、制御対象を希望するように動作させることを( **制御** )といいます。

制御対象(モータ)



【動作】



### 手動制御と自動制御

人がスイッチを操作する( **手動制御** )の代わりに、時計やサーモスタットなどの機械やコンピュータで自動的に制御することを( **自動制御** )といいます。

#### 機械による自動制御の例

- ・こたつ…冷たくなれば、電流を流して温度を上げる (サーモスタット、ヒータ)
- ・スプリンクラー…時間がくれば、一定時間水をまく (時計、ポンプ)



### コンピュータ制御

最近の自動制御では( **コンピュータ制御** )が多くなっています。

身の回りのコンピュータ制御の多くは( **マイコン** )と呼ばれる小さなコンピュータによって制御されています。



## 2. 計測・制御システム

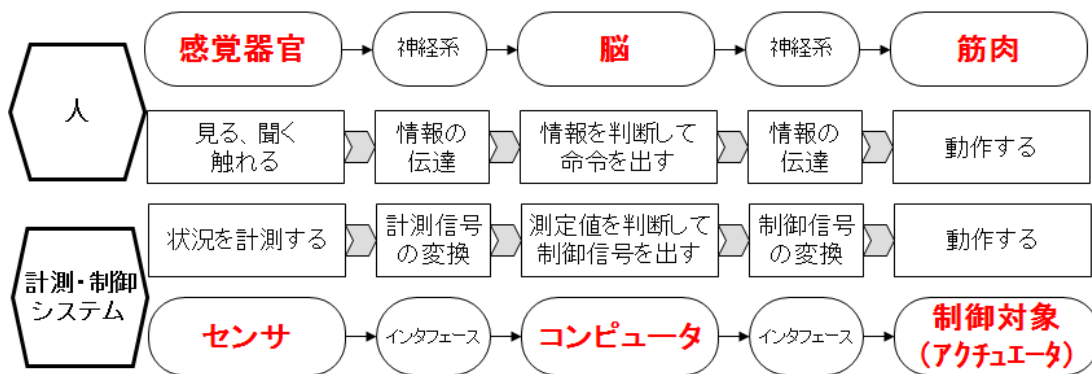
### 入力と出力

コンピュータに信号を送ることを( **入力** : Input )するといひ、コンピュータから信号を取り出すことを( **出力** : Output )するといひます。

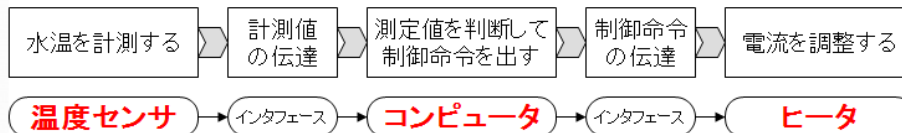
コンピュータ制御では、人の感覚器官のかわりとなる( **センサ** )からの信号を入力し、LED やモータなどの( **制御対象** )を動作させる信号(制御信号)を出力します。

### 計測・制御システム

コンピュータ制御する一連の機械やソフトウェアの集まりを( **計測・制御システム** )と呼びます。



#### 電気ポットの計測・制御例



### アクチュエータとは

「動作させるもの」を意味し、モータや油圧シリンダなど、与えられたエネルギーで機械的な動作をする部品を( **アクチュエータ** )といひます。

### インタフェースとは

接触面、境界面の意。機械やソフトウェア、人間などの二つの異なるものを接続する機能や規格のことを( **インタフェース** )といひ、お互いの橋渡しの役割をします。

センサで計測された情報は、インタフェースでデジタル化され、コンピュータに送られます。またコンピュータの命令は、制御対象が理解できる電気信号などにインタフェースで変換されます。

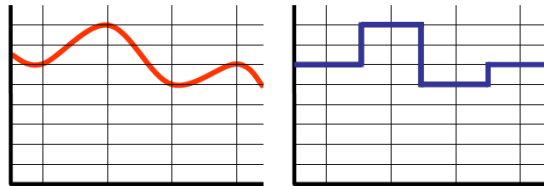
## センサの種類

機械や環境の状態を計測するセンサには、いろいろな種類のものがあります。

光を検知する	( <b>CdS セル</b> )	明るさにより抵抗値が変化するセンサ
	フォトダイオード	光エネルギーを電気エネルギーに変換するセンサ
	( <b>赤外線センサ</b> )	赤外線（人の目には見えない）を検知するセンサ
	( <b>CCD</b> )	多くのフォトダイオードを並べて、画像を取得できるようにしたセンサ Charge Coupled Device
温度を感知する	( <b>サーミスタ</b> )	温度の変化で抵抗値が変化するセンサ。
磁気を検知する	磁気センサ	テープ上の磁気信号を電気信号に変換するセンサ。
力を検知する	マイクロスイッチ	軽くふれると ON-OFF できるスイッチ。
	感圧センサ	圧力が加わると抵抗が変化するセンサ。 ※自動車の乗員着座感知など

## アナログとデジタル

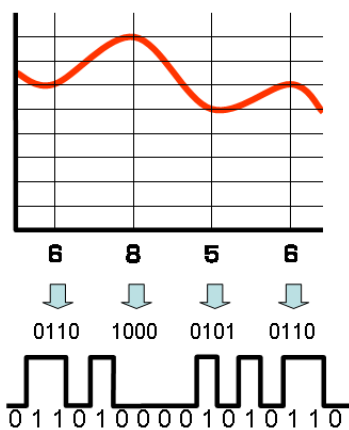
センサから得られるデータは、なめらかに連続した変化を示す( **アナログ** )信号と、スイッチの ON-OFF のように階段状に変化する( **デジタル** )信号があります。



## A/D変換

アナログ信号は、コンピュータで処理しやすいようにデジタル信号に( **A/D変換** )します。

またコンピュータで処理されたデジタル信号は、これに応じた電圧や電流に変えて利用することがあり、これをする装置のことを( **D/A変換器** )といいます。



### 量子化

例えば0Vから5Vの電圧を4ビットで処理するAD変換では、測定値を0から15の整数に変換します。これを**量子化**といいます。

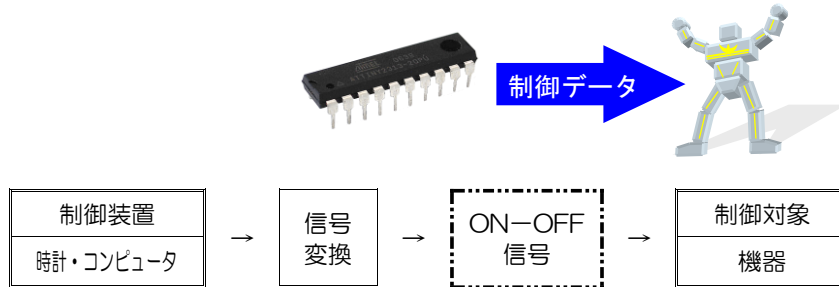
0	～	0.3125V	→	0
0.3125	～	0.625V	→	1
.....				
4.375	～	4.6875V	→	14
4.6875	～	5V	→	15

※量子化するビット数が大きいほど精度が高くなります。

### 3. 制御の種類

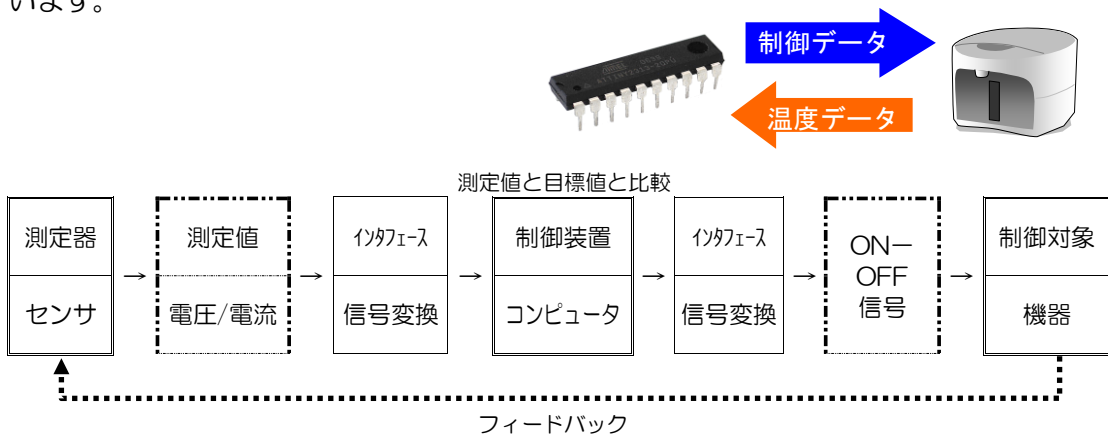
#### シーケンス制御

あらかじめ定められた( **手順** )に従って工程を進めて行く制御を( **シーケンス制御** )といい、開ループ制御ともいいます。



#### フィードバック制御

( **センサ** )から送られてきた温度や明るさなどの( **測定値** )を( **目標値** )に近づけるようにする制御を( **フィードバック制御** )といい、閉ループ制御ともいいます。



#### 身の回りにおける制御

家庭や社会にある機械では、フィードバックのある制御がたくみに行われています。

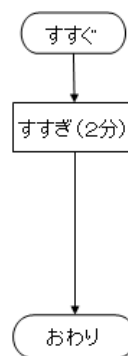
※全自動洗濯機の例

- (1) スタートボタンを押す (2) 洗い
- (3)すすぎ (4)脱水 (5) 終了

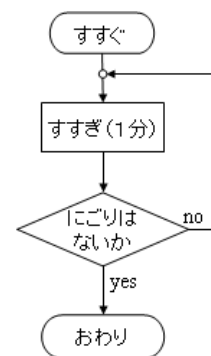
という決められた手順でシーケンス制御されています。

しかし、例えば「すすぎ」の工程では、汚れセンサでにごりの測定値を得て、一定の目標値に近づけるまですすぎを繰り返すフィードバック制御が行われています。

#### シーケンス制御



#### フィードバックのある制御



## 4. アルゴリズムとフローチャート

### アルゴリズム

問題を解くためには、まず手順を考えます。この手順を( **アルゴリズム** )といいます。

このアルゴリズムを理解しやすくするために( **フローチャート** ) (流れ図) で表します。

#### 問題解決までの流れ

アルゴリズムを考える： 問題の解決までの手順  
↓  
フローチャートの作成： アルゴリズムを図示する  
↓  
プログラムの作成： プログラム言語で記述  
↓  
プログラムの実行



### 順次、分岐、くり返し(反復)処理

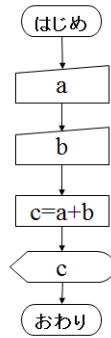
問題解決のための基本的な処理には、

- ( **順次** ) 処理、
- ( **くり返し** ) 処理、
- ( **分岐** ) 処理

があります。

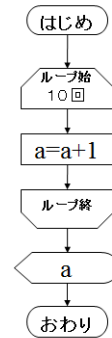
フローチャートでは、右の図のように表現できます。

#### 順次処理



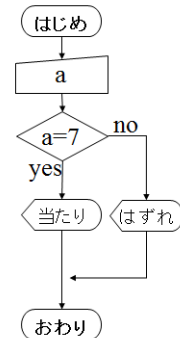
aとbの値を入力して  
合計の値を表示する

#### くり返し処理



1を10回足して  
合計の値を表示する

#### 分岐処理



aの値を入力して、  
7の時は「当たり」  
それ以外の時は「外れ」  
を表示する

### フローチャートの図記号

フローチャートには、処理を表す図記号やその流れを示す矢印などが使われています。

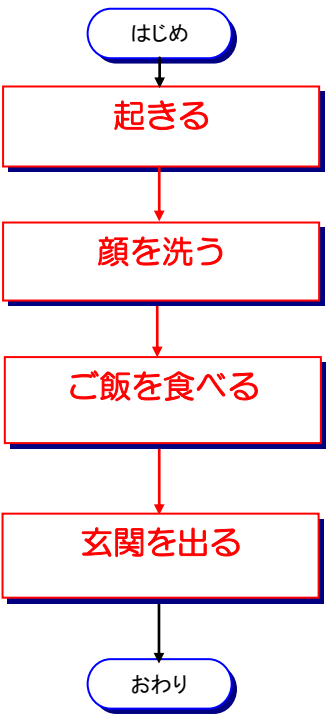
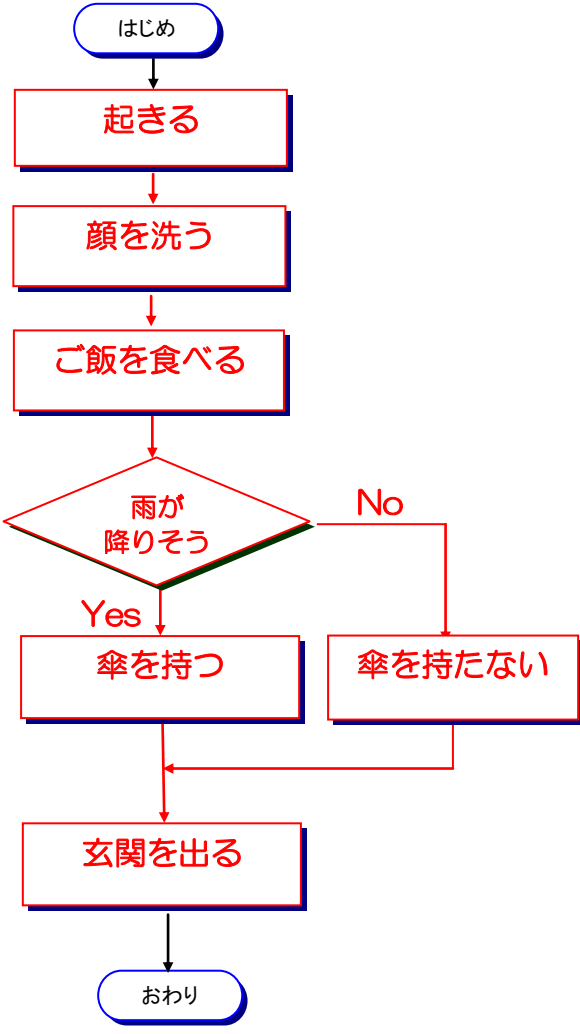
	<b>端子</b>	処理の開始・終了		<b>手操作入力</b>	キーボードからのデータの入力
	準備	変数の宣言、初期値の設定など		<b>表示</b>	ディスプレイでのデータの表示
	<b>処理</b>	判断などの処理以外の処理		書類	データの印刷
	サブルーチン	あらかじめ定義された処理		<b>判断</b>	条件により、流れが二つ以上に分岐する
	入出力	データの入出力		<b>ループ端</b>	くり返しの開始と終了



## フローチャートを書こう

### 課題

朝目覚めてから、家を出るまでをフローチャートで書こう

<p>(例) 朝起きて、顔を洗って、ご飯を食べて、玄関を出るまでの行動をフローチャートに書いてみよう</p>	<p>(例) 朝起きて、顔を洗って、ご飯を食べて、傘を持つかどうかを判断して、玄関を出るまでの行動をフローチャートに書いてみよう。</p>
 <pre>graph TD; A([はじめ]) --&gt; B[起きる]; B --&gt; C[顔を洗う]; C --&gt; D[ご飯を食べる]; D --&gt; E[玄関を出る]; E --&gt; F([おわり]);</pre>	 <pre>graph TD; A([はじめ]) --&gt; B[起きる]; B --&gt; C[顔を洗う]; C --&gt; D[ご飯を食べる]; D --&gt; E{雨が降りそう?}; E -- Yes --&gt; F[傘を持つ]; E -- No --&gt; G[傘を持たない]; F --&gt; H[玄関を出る]; G --&gt; H; H --&gt; I([おわり]);</pre>

## 5. プログラムとプログラム言語

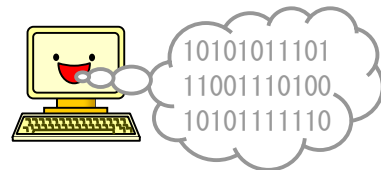
### プログラム

コンピュータに仕事をさせるための一連の手順を記述したものを（ **プログラム** ）といいます。また、これを作成することをプログラミング、作成する人をプログラマといいます。

### プログラム言語

○プログラムを書くための言葉を（ **プログラム言語** ）といいます。

○プログラム言語は、プログラムの目的によって色々ありますが、コンピュータが理解できる言語は2進数で書かれた（ **機械語** ）だけです。これを書かれたプログラムを理解できる人はとても限られています。



○機械語に対して、多くの人が理解しやすいプログラム言語を（ **高級言語** ）または高水準言語といい、（ **BASIC** ）や（ **C言語** ）（ **JAVA** ）などがあります。

```
:1000000012C060C229C028C027C072C025C024C049  
:1000100023C022C021C020C01FC01EC01DC01CC0E4  
:100020001BC01AC019C011241FBECFEDCDBF10E0F8
```

機械語の例（2進数を16進数に変えた形で表示）

```
Do  
  Waitms 100  
  Gosub Mplay  
Loop  
End
```

高級言語の例(BASIC)

#### 機械語

コンピュータが理解できる唯一の言語。マシン語とも呼ばれている。実行速度は速いのですが、「0」と「1」だけの2進数で書かれているため、普通の人には理解できません。

#### BASIC

BASICは（ **初心者** ）向けの言語として開発されましたが、現在は広くプログラマに使われています。

Beginner's All-purpose Symbolic Instruction Code の略

#### C言語

OSやアプリケーションの開発に広く使われている言語です。

#### JAVA

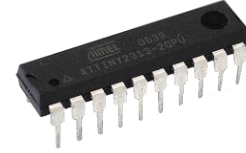
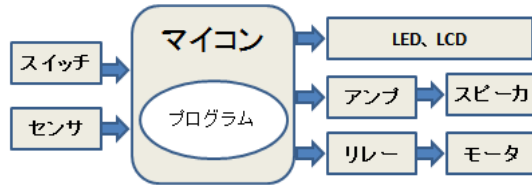
Java は OS やコンピュータを問わずに動作するという利点があります。Web 系にも強いなどから、Android 携帯などのアプリケーションの開発にも使われることが多くなっています。



## 6. マイコンとプログラム

### マイコンとパソコン

マイコンは( **プログラム** )を書き込んで利用する小さな( **コンピュータ** )です。



### ピンとポート 参照→p.33

マイコンには( **ピン** )と呼ばれる足が数本から数10本あります。このうちのいくつかのピンを通じて( **データ** )の入出力を行います。

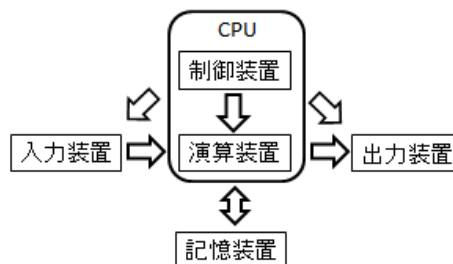
マイコンのデータを入出力するピンのいくつかをまとめて( **ポート** )といいます。

#### マイコンとは

マイクロコンピュータまたはマイクロコントローラの略。コンピュータが持つ構成要素をICチップ(LSI)上に搭載したもので、ワンチップマイコンとも呼ばれる。小さな部品で特定機能の処理を行うことができるため、家電製品や電子機器などのコンピュータ制御を必要とする装置に組み込まれている。マイコンという言葉は和製英語。英語では「microcomputer」「microcontroller」または略して「micro」と表現される。

#### コンピュータの持つ5つの構成要素

要素	パソコン	マイコン
制御装置	CPU	CPU
演算装置		
記憶装置	RAM、HDD、SSD	RAM
入力装置	キーボード、マウス	スイッチ、センサ(光、音、熱など)
出力装置	モニタ	LED、LCD など



## マイコンとプログラム言語

マイコンに書き込みできるプログラムは、( **機械語** ) で書かれたものだけです。BASICなどの、機械語以外の言語で書かれたプログラムは、機械語に変換(翻訳)する必要があります。このように異なる言語に変換することを( **コンパイル** )といい、またコンパイルするためのソフトウェアをコンパイラといいます。

※コンパイル前に作成されたプログラムをソースファイルと呼ぶことがあります。

## プログラムの書き込み

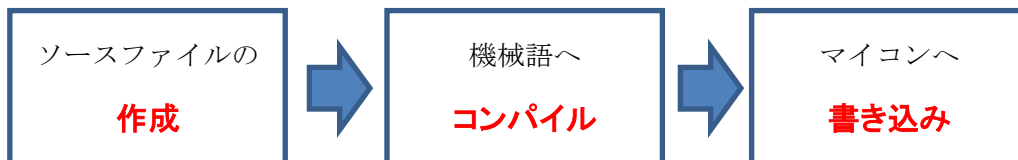
プログラムをマイコンに書き込む道具を( **ライター** )または**プログラマ**といいます。

- ※ ヒダピオ学習回路は、スイッチを上にしてUSBポートにさし直すとライターになります。(右から3番目のLEDが点灯します)



## プログラムの作成から書き込みまで

プログラムをマイコンに書き込むための手順は



【参考】「第3章プログラムの作成」で使用するソフトウェアとプログラム言語の関係は

作業の内容	使用するソフト	プログラム言語
①ソースファイルの作成	ArduBlock	グラフィック言語
②コンパイル		→ C言語(Arduino言語)
③マイコンへ書き込み	Arduino IDE	C言語(Arduino言語)
	hidspix	→ 機械語



## 7. 第1章のまとめ



### 自己評価

計測・制御システムの基本的なしくみが理解できたか。	A B C D
センサの種類や信号変換が理解できたか。	A B C D
フィードバック制御について理解できたか。	A B C D
家庭や社会にある機械の制御例について理解できたか	A B C D
問題解決までの手順が理解できたか。	A B C D
順次、繰り返し、分岐処理について理解できたか。	A B C D
フローチャートを書くことができたか	A B C D
プログラム言語とマイコンの関係について理解できたか。	A B C D
熱心に授業に取り組めたか。	A B C D

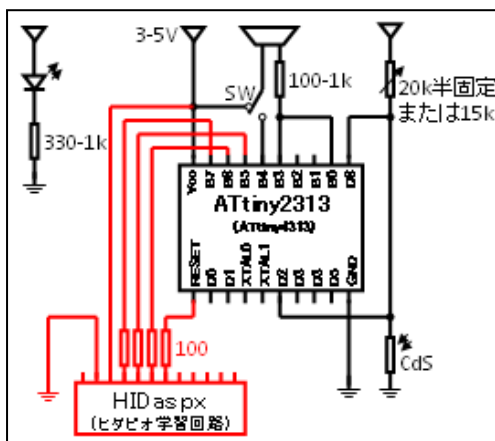
### 感想

これまで学習して感動したことや、疑問に思って解決できたこと、将来に役立ちそうなことなどを中心に、感想にまとめよう。

## 第2章 時計付き電子オルゴールの製作

### 1. 概要

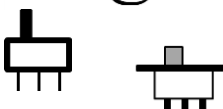
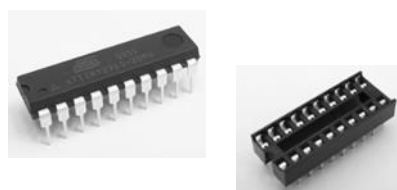
AVR マイコンを使った電子回路を作り、好きな曲をマイコンに書き込み、センサを使って音楽を鳴らします。



電子オルゴールの回路図

### 2. 部品一覧

- |                       |                          |
|-----------------------|--------------------------|
| (1) スピーカ              | <input type="checkbox"/> |
| (2) 時計                | <input type="checkbox"/> |
| (3) AVRマイコン           | <input type="checkbox"/> |
| (ATtiny2313 または 4313) |                          |
| (4) ICソケット 20ピン       | <input type="checkbox"/> |
| (5) 抵抗器 100Ω (茶黒茶金)   | <input type="checkbox"/> |
| 680Ω (青灰茶金)           | <input type="checkbox"/> |
| 15kΩ (茶緑橙金)           | <input type="checkbox"/> |
| (6) CdSセル             | <input type="checkbox"/> |
| (7) スライドスイッチ 2個       | <input type="checkbox"/> |
| (8) 電源用スライドスイッチ       | <input type="checkbox"/> |
| (9) LED               | <input type="checkbox"/> |
| (10) I型ピンヘッド 1×6      | <input type="checkbox"/> |
| (11) 電池ケース            | <input type="checkbox"/> |
| 電池 2本                 | <input type="checkbox"/> |
| (12) リード線 緑 2本        | <input type="checkbox"/> |
| (13) はんだ              | <input type="checkbox"/> |
| (14) ボックス用整形木板 13枚    | <input type="checkbox"/> |
| (15) 時計用針 3本          | <input type="checkbox"/> |
| (16) 時計用ナット           | <input type="checkbox"/> |



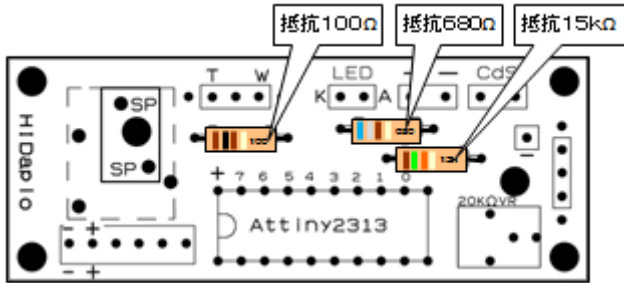
特に小さい部品は  
なくさないようにしましょう！



### 3. 部品のはんだ付け

(1) 抵抗器 (100Ω、680Ω、15kΩ) をはんだ付けする。

※はんだ付け後は、余分なリード線を切り取りましょう。



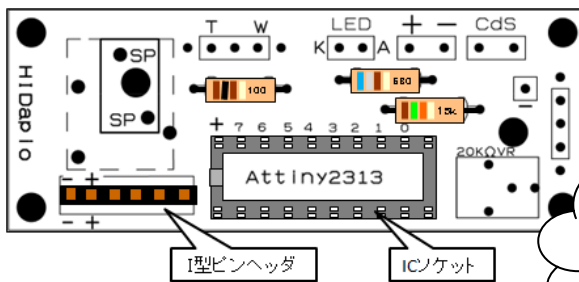
抵抗は、カラーコードを確認してください！



【参考】

数値	0	1	2	3	4	5	6	7	8	9
色	黒	茶	赤	橙	黄	緑	青	紫	灰	白

(2) I型ピンヘッド、ICソケットをはんだ付けする。

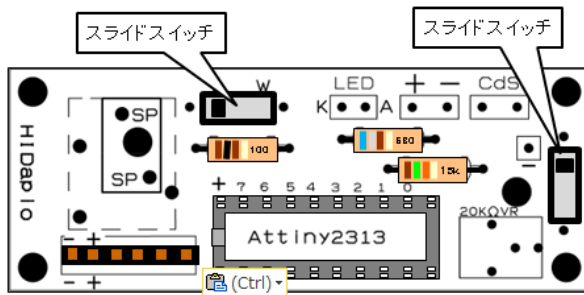


ICソケットの足は、全てきちんと穴に入れます！

1つのピンをはんだ付けた後、ゆがみがないか確かめよう！



(3) スライドスイッチを2個はんだ付けする。

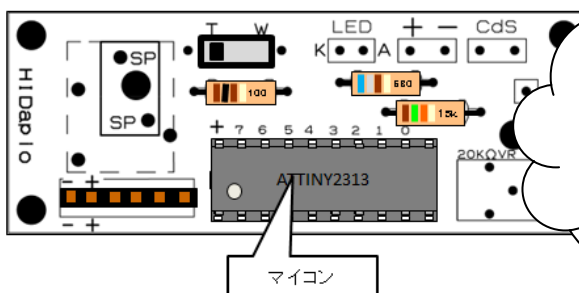


1本の足をはんだ付けた後、ゆがみがないか確かめよう！



(4) マイコンを取り付ける。

※マイコンは、ATtiny2313 または ATtiny4313 を使います。



マイコンは、向きを間違えないようにして、全ての足を IC ソケットに入れます！



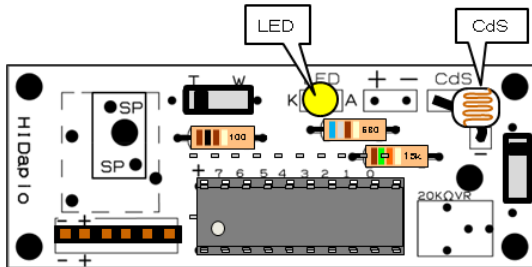
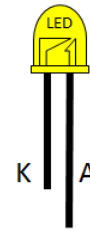
(5) LED とCdS をはんだ付けする。

○LED は長い方の足を「A (アノード)」側に、

短い方の足を「K (カソード)」側に取り付ける。

○CdS の下図をよく見て、足をさす場所を間違えないようにする。

※はんだ付け後は、余分なリード線を切り取りましょう。



CdS の足はどちらをさしてもいいけど、片方の足のさす場所に気をつけて！

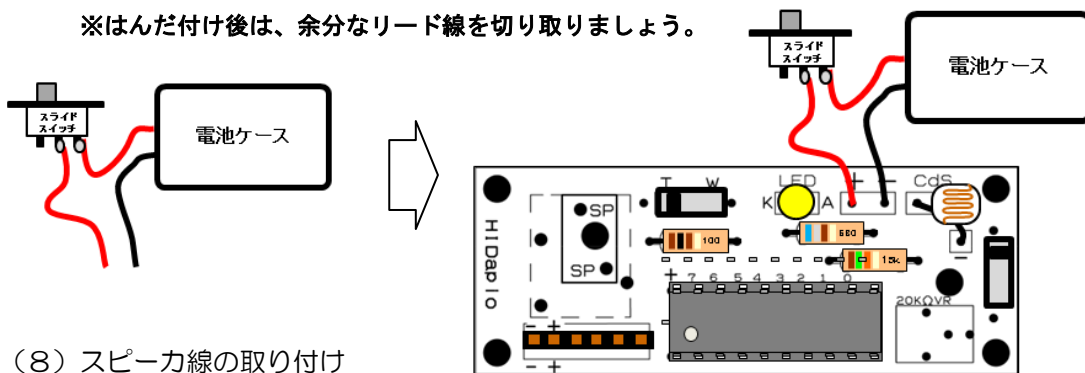


(7) 電源線の取り付け

○スライドスイッチに電池ケースの赤 (+) の線と、赤のリード線をはんだ付けする。

○赤のリード線を基板の「+」に、電池ケースの黒の線を「-」にさし、はんだ付けする。

※はんだ付け後は、余分なリード線を切り取りましょう。

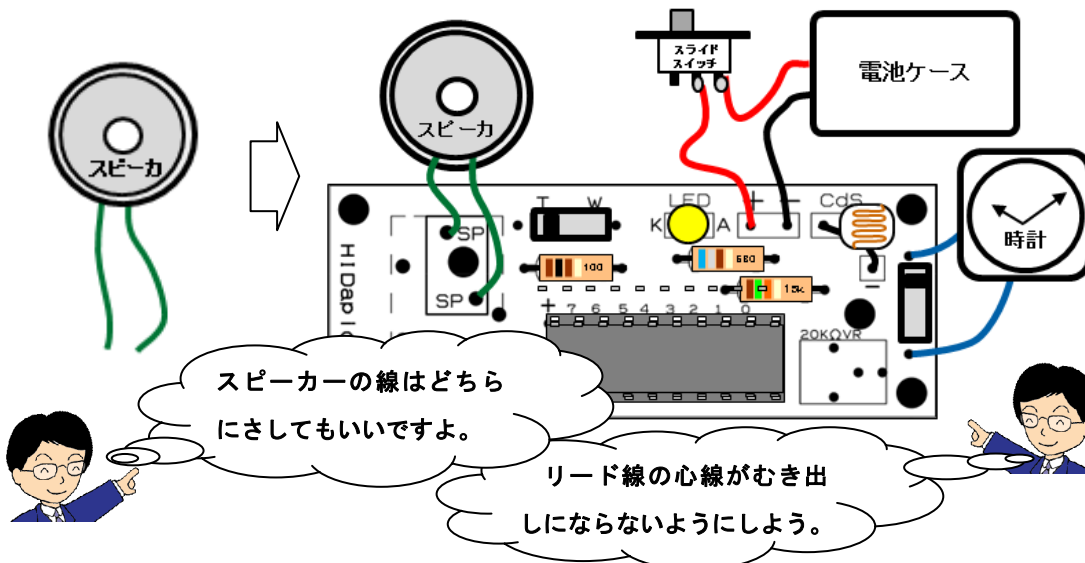


(8) スピーカ線の取り付け

○スピーカに緑のリード線を2本はんだ付けする。

○2本のリード線を基板の「SP」穴にさし、はんだ付けする

※はんだ付け後は、余分なリード線を切り取りましょう。



スピーカの線はどちらにさしてもいいですよ。

リード線の心線がむき出しにならないようにしましょう。



## 4. ボックスの仮組み立て

- (1) ボックス用整形木板を説明書を見ながら、ていねいに仮に組み立てましょう。  
○組みあわせが合わない場合は、カッターナイフで少しだけ削ると上手く組み合わせることができます。※ゆるすぎないように、削りすぎに注意！
- (2) 仕上げ削り、塗装などをします。

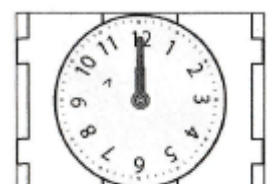
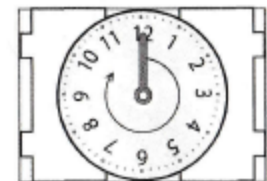
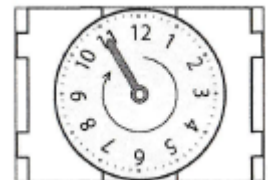
## 5. 部品の取り付け

- (1) 基板とスライドスイッチをねじで取り付けます。
- (2) スピーカーをねじで取り付けます
- (3) 時計ケースを取り付け、文字盤シールを張ります。



## 6. 時計針の組み立て

- (1) 長針だけをはめて、カチッと音がするところまで回す。
- (2) 長針をそのまま抜いて、12時を指すようにはめ直す。
- (3) 長針を回して、12時でカチッと音がするか確認する。  
※ずれているときは(1)からやり直す。
- (4) 長針を抜く。※回さないように注意しよう
- (5) 短針、長針、秒針の順に  
12時を指すように、順番に重ねてはめる。



## 7. 第2章のまとめ

### 自己評価

部品の確認ができ、名称を覚えられたか。	A B C D
部品を、図面通りに取り付けることができたか。	A B C D
はんだ付けが、手際よく上手くできたか。	A B C D
余分な線が無いように、処理ができたか。	A B C D
組み立ては、無理なくできたか。	A B C D
ケガをしたり、しそうなになったりすることが無かったか。	A B C D
熱心に授業に取り組めたか。	A B C D

### 感想

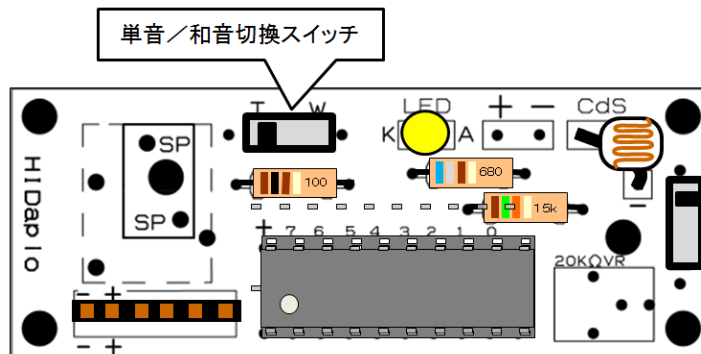
これまで学習して感動したことや、疑問に思っ解決できたこと、将来に役立ちそうなことなどを中心に、感想にまとめよう。

## 第3章 プログラムによる制御

### 1. 制御学習の準備をしよう

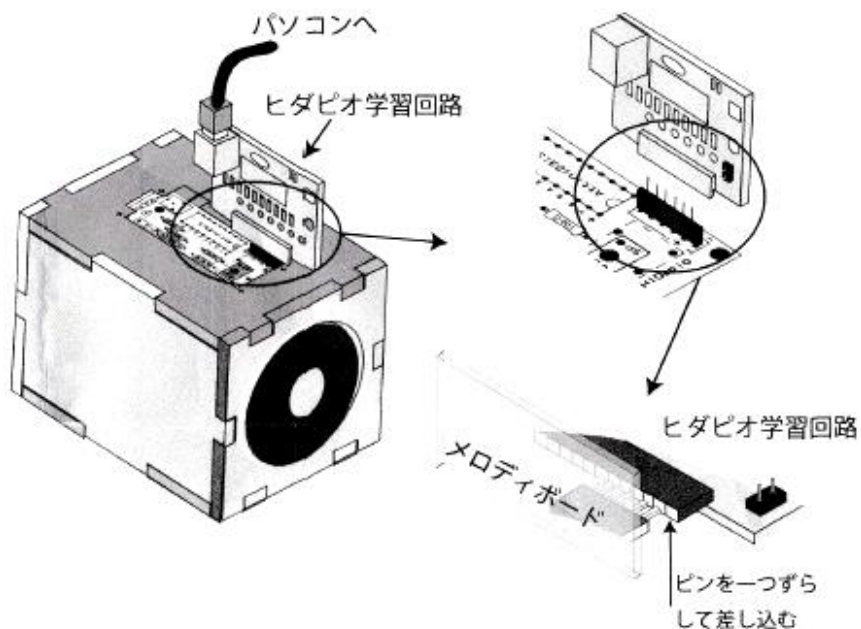
#### 電子オルゴールのスイッチの設定

基板の和音／単音スイッチを単音側（左側）にします。



#### ヒダピオ学習回路、電子オルゴールを接続

ヒダピオ学習回路のスイッチを上にしてから、電子オルゴールを接続し、パソコンのUSBポートに接続します。



USBケーブルをUSBポートにさしたときに、「ピポツ」と音がすれば正常に動作していると判断できます。

※接続時に音がしないときは、パソコンのボリュームが小さくなっていないかチェックしよう。



## 2. アルドゥブロックの基礎

### ArduBlock(アルドゥブロック)を起動する

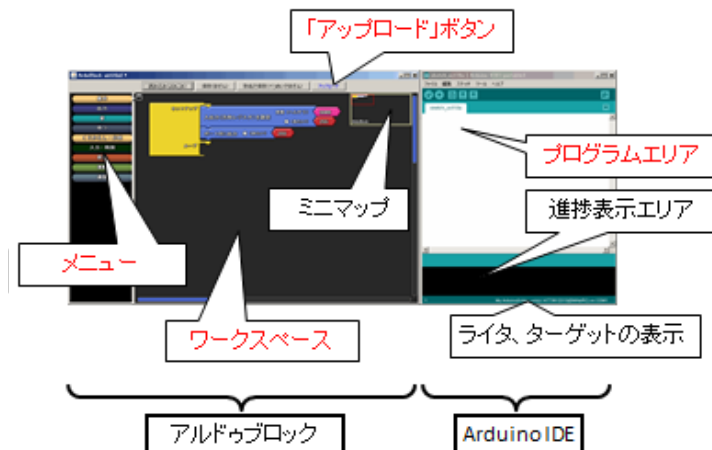
参照→p1,2

①Arduino IDE (アルディーノ アイディイー) を起動します。

② arduino.exe をクリックする。

③ 「ツール」 → 「ArduBlock」 で、学習を開始します。

### アルドゥブロックの画面の構成



### 初めて使用するときは

①初めて使用する時は、( **マイコンとライタを指定** ) します。

- 「ツール」 → 「マイコンボード」 で「ATtiny2313」  
※マイコンに「ATtiny4313」を利用するときは「ATtiny4313」
- 「ツール」 → 「書込装置」は「HIDasp (トップマン製ヒダピオ学習回路)」  
※書込装置に自作の HIDasp を使用する場合は「HIDasp」

②続いて ( **マイコンの設定** ) を書き変えます。

- 「ツール」 → 「ブートローダを書き込む」で、マイコンのヒューズ設定を書き変えます。

```
Detected device is ATtiny2313.  
Lock bits are programmed (0xCF).
```

左のメッセージが  
進捗表示エリアに表示  
されたら OK!



## プログラムの作成

- ①「メニューボタン」からブロックをドラッグ&ドロップしてプログラムを作成します。
- ②「セットアップ・ループ」ブロックは一つだけを必ず配置します。
- ③ブロックのほとんどは「ループ」の中に配置します。
- ④不要なブロックは、画面の左側にドラッグ&ドロップで消去します。
- ⑤ブロックの上で右クリックすると、ブロックをコピーできます。

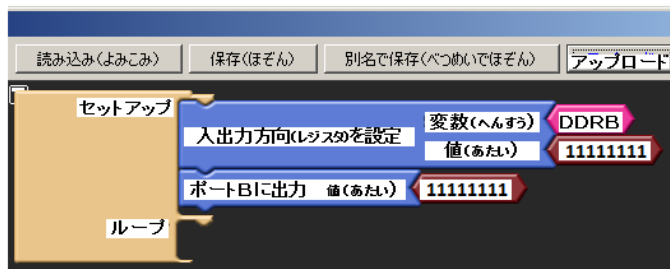
## プログラムの書き込み

アルドゥブロックでブロック形式のプログラムを作成したら、「アップロード」ボタンをクリックします。すると、

- ①C言語（Arduino言語）のスケッチ（プログラム）にコンパイルされ、プログラムエリアに表示されます。

※Arduinoでは、プログラムのことをスケッチと呼んでいます。

スケッチを直接操作して、書き込むこともできます。



```
sketch_feb09a | Arduino 0101-pc
ファイル 編集 スケッチ ツール ヘルプ
sketch_feb09a $
/* This is the start */
void mysetup();

void setup()
{
  mysetup();
}

void mysetup() {
  DDRB = B11111111 ;
  PORTB = B11111111 ;
}
```

- ②スケッチ（プログラム）は機械語にコンパイルされてマイコンに書き込まれます。

```
Passed.
Total read/write size = 780 B / 0.53 s (1.44 kB/s)
```

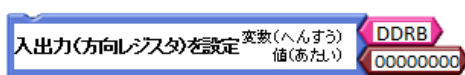
左のようなメッセージが  
進捗表示エリアに表示  
されたらOK!



## 3. 主なブロックのはたらき

### セットアップと入出力の設定

プログラムの初めのセットアップ部に（ **入出力の設定** ）をします。



- 入出力(方向レジスタ)の設定  
ポートBの8つのピンが、「1」の時は出力ピン、「0」の時は入力ピンになります。  
「11111111」でB0~B7の全てのピンが( **出力** )ピンになります。

## 出力ブロック

マイコンにある各ピンの出力値の設定をするために、  
( **出力** )ブロックがあります。

出力ブロックには、すべてのピンの出力値を一つ一つ設定するものと、ポート B の全てのピンの出力値を設定するものがあります

ピンの出力値は、次の命令があるまで変わりません。



ATtiny2313 のピン

	<p>ピン# (ナンバー) と値を設定 「PB0」でポートBの0番ピンを 「LOW」で (出力あり) に、 「HIGH」で (出力なし) にします。</p>
	<p>値はPB7～PB0を一度に設定する。 「0」で「LOW」(出力あり) 「1」で「HIGH」(出力なし) とピンに出力します。</p> <p>「11111111」で全てのピンで出力なし 「00000000」で全てのピンで出力あり 「11111110」でB0ピンだけ出力あり</p>

### ※エラーメッセージとその対応(1)

`HIDaspx-p(*) isn't found.` …「ヒダピオ学習回路」が接続されていません。

→ 「ヒダピオ学習回路」を USB ポートに接続する。

`HIDaspx(*) isn't found.` …書込装置が「HIDaspx」になっています。

→ 「ツール」→「書込装置」を「HIDaspx(トップマン製ヒダピオ学習回路)」にする。

`Device connection failed.(PE)` …マイコンがつながっていません。

→ 「ヒダピオ学習回路」にオルゴールを接続する。

`Warning: Please check HIDaspx mode.  
Detected device is ATtiny2313.  
Lock bits are programmed (0xCF).` …ヒダピオ学習回路が USB-IO モードです。

→ 「ヒダピオ学習回路」のモード切換スイッチを切り換え、USB ポートに再接続する。



## 4. プログラムを組もう

### くり返し(反復)処理

プログラムの中で、同じことを繰り返し行うことを( **くり返し** )処理または反復処理あるいはループ処理といいます。



### 無限くり返し

くり返し処理の中で、永遠にくり返す処理を( **無限くり返し** )といいます。

### 実習1

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

※スピーカに耳を当てて、音を確認めよう

	プログラム	意味
ループ	ポートBに出力 値(あたい) 11111111	PBOピン「1」=HIGH=出力なし
	待つ (まつ) 長さ(ミリ秒) 1000	待つ 1000m秒=1秒
	ポートBに出力 値(あたい) 11111110	PBOピン「0」=LOW=出力あり
	待つ (まつ) 長さ(ミリ秒) 1000	待つ 1000m秒=1秒
結果	「ブッ ブッ ブッ」と断続音が続く。	

### 実習2

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

	プログラム	意味
ループ	ポートBに出力 値(あたい) 11111111	PBOピン「1」=HIGH=出力なし
	待つ (まつ) 長さ(ミリ秒) 100	待つ 100m秒=0.1秒
	ポートBに出力 値(あたい) 11111110	PBOピン「0」=LOW=出力あり
	待つ (まつ) 長さ(ミリ秒) 100	待つ 100m秒=0.1秒
結果	「ブッブッ」の連続音が続く。	

### 実習3

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

	プログラム	意味
ループ	ポートBに出力 値(あたい) 11111111	PBOピン「1」=HIGH=出力なし
	待つ (まつ) 長さ(ミリ秒) 10	待つ 10m秒=0.01秒
	ポートBに出力 値(あたい) 11111110	PBOピン「0」=LOW=出力あり
	待つ (まつ) 長さ(ミリ秒) 10	待つ 10m秒=0.01秒
結果	「ブー」という低い連続音が続く。	

## 考察

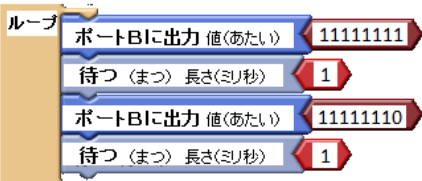
実習1～3の結果から、どうしてこのような音が鳴るのだろうか？

考察

出力の「あり」と「なし」の入れ替わるときに音がするから。

## 実習4

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

プログラム	意味
	PBOピン「1」=HIGH=出力なし 待つ 1m秒=0.001秒 PBOピン「0」=LOW=出力あり 待つ 1m秒=0.001秒

結果

「プー」という高い連続音が続く。

## 課題1

待つ長さを変えて、いろんな音を確認しよう

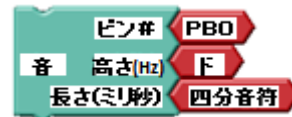
結果

と感想

## 課題2

音階を入れて、チャルメラを作ろう

「ヒント」ドレミーレド、ドレミレドレー



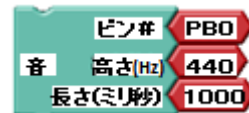
結果

と感想

## 課題3

救急車の音を再現しよう。

「ヒント」救急車のサイレン音は、770Hzと960Hz



結果

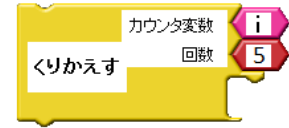
と感想



## 有限くり返し

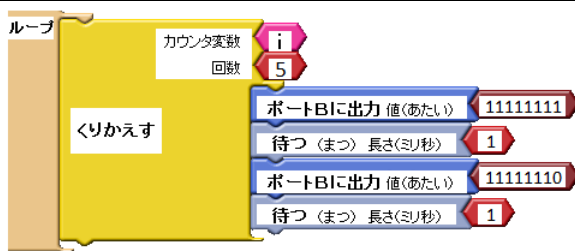
プログラムを、決められた回数だけくり返すことを( **有限くり返し** )処理といいます。

例えば、右のブロックの場合、i の値が1 から5 まで変化するまで、5 回くり返します。この i のように値が変化するものを( **変数** )といいます。



### 実習1

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

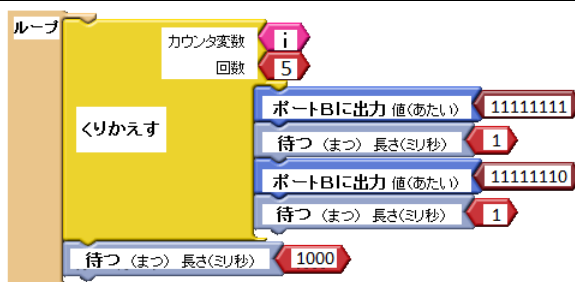


5 回くり返し=「プツという音」を無限にくり返す。

結果 「プー」という高い連続音が続く。

### 実習2

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

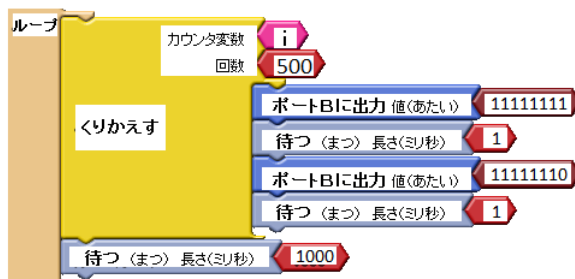


5 回くり返し=「プツという音」  
待つ 1000ms=1 秒  
を無限にくり返す。

結果 「プツ プツ」という高い断続音が続く。

### 実習3

次のプログラムをループの中に組み、「アップロード」ボタンを押そう

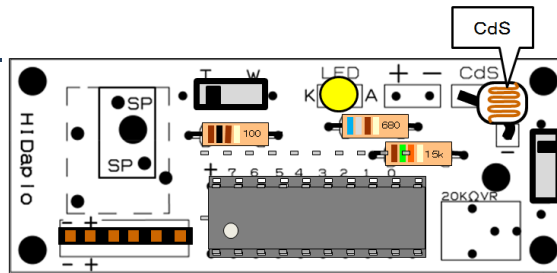


500回くり返し=「プーという音」  
待つ 1000ms=1 秒  
を無限にくり返す。

結果 「プー プー」という高い断続音が続く。

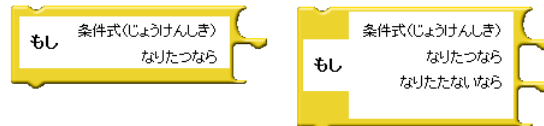
## 5. センサを使おう

### センサ (CdS セル) を使おう



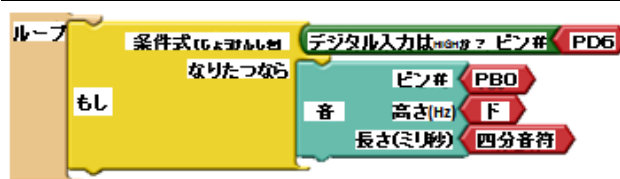
### 入力によって動作を変えるプログラム (条件分岐)

プログラム中で、ある条件が満たされているかどうかによって次に実行するプログラムを切り替える命令を( **条件分岐** )処理といいます。



### 実習1

次のプログラムをループの中に組み、「アップロード」ボタンを押そう



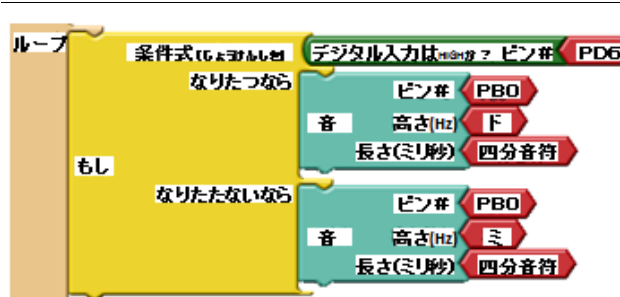
ポートDの6番ピンが HIGH=入力なし(=暗い)ならば  
ドの音が鳴る  
を無限にくり返す。

結果 **暗くなると「ド」が鳴る。**

※プログラムをアップロードして CdS セルに手をかざすと、音が鳴ります。

### 実習2

次のプログラムをループの中に組み、「アップロード」ボタンを押そう



ポートDの6番ピンが HIGH=入力なし(=暗い)ならば  
ドの音が鳴る  
そうでなければ(=明るい)  
ミの音が鳴る  
を無限にくり返す

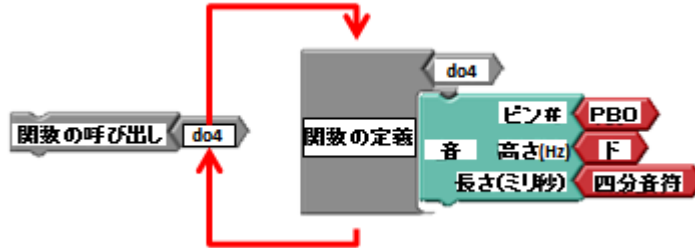
結果 **暗いと「ド」が鳴り、明るい「ミ」が鳴る。**

※プログラムをアップロードして CdS セルに手をかざすと、音が変わります。

## 6. 関数(サブルーチン)を使おう

### 関数とは

プログラム中で、一連の処理をまとめて一つの手続きとしたものを( **関数** )といい、同じ処理をくりかえす場合などに使用されます。プログラム言語によってはサブルーチンなどと表現されます。



### 実習1

次のプログラムをループの中に組み、「アップロード」ボタンを押そう



ポート D の 6 番ピンが

HIGH=入力なし (=暗い) ならば ドの音が鳴る

そうでなければ (=明るい) ミの音が鳴る

を無限にくり返す

### 課題 1

関数の中身を変えて、いろんな音を確認しよう

結果  
と感想

## 7. プログラム総合実習

### 課題

自由にプログラムを考えてみよう。

z※画面をハードコピーし、印刷して貼り付けよう。

結果  
と感想

#### 画面のハードコピーの方法

- ① PrintScreen キー（機種によっては Fn キー+PrintScreen キー）を押す。  
※キーボードの表示が「PrtScr」などに鳴っている機種もある  
※現在のウィンドウだけをハードコピーする場合は Alt キーも同時に押す。
- ② 右クリックして「貼り付け」する。



## 第3章のまとめ



### 【自己評価】

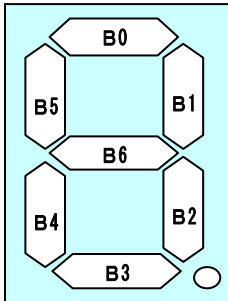
実験装置が正しく接続できたか。	A B C D
ソフトウェア「ArduBlock」を操作できたか。	A B C D
マイコンの設定の意味を理解し、設定できたか。	A B C D
入出力設定の意味が理解できたか。	A B C D
くり返し（無限、有限）のプログラムが組めたか。	A B C D
変数の役割とはたらきが理解できたか。	A B C D
センサのはたらきが理解できたか	A B C D
条件分岐のプログラムを理解できたか	A B C D
熱心に授業に取り組めたか	A B C D

### 感想

これまで学習して感動したことや、疑問に思っ解決できたこと、将来に役立ちそうなことなどを中心に、感想にまとめよう。

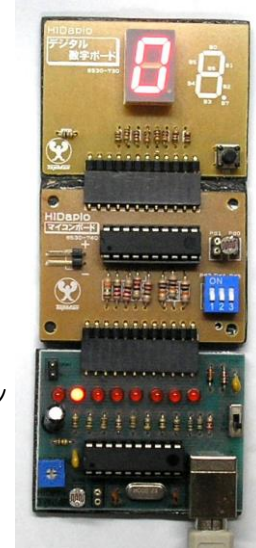
# 第4章 いろいろな制御

## 1. デジタル数字



7セグメントLEDの各セグメントは、図のようにB0～B6ピンにつながっています。

※右下の点は、B7ピンにつながっています



### プログラムの作成から動作まで

「ArduBlock」でプログラムを作成する。

(例) 数字「1」を1秒間点灯する



(解説) B1とB2を「点灯」=「0」にする。

### 課題

デジタル数字の1～0を表してみよう

(1) 下の図の各セグメントに色を塗って、出力指定数を【 】内に記入しよう。

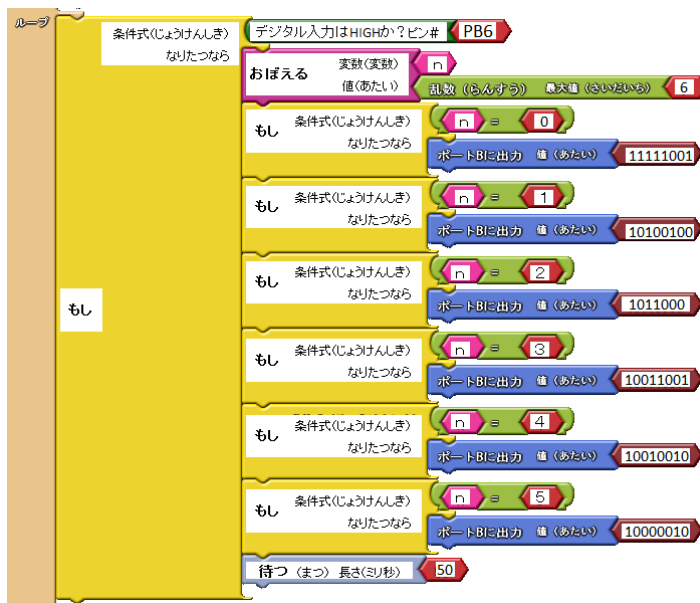
数字	1	2	3	4	5
数字	6	7	8	9	0

(2) 1～0までの数字を表示するプログラムを組もう。

動作		ポートBの値
数字表示	B6 B5 B4 B3 B2 B1 B0	
1		1 1 1 1 1 0 0 1
2		1 0 1 0 0 1 0 0
3		1 0 1 1 0 0 0 0
4		1 0 0 1 1 0 0 1
5		1 0 0 1 0 0 1 0
6		1 0 0 0 0 0 1 0
7		1 1 1 1 1 0 0 0
8		1 0 0 0 0 0 0 0
9		1 0 0 1 0 0 0 0
0		1 1 0 0 0 0 0 0

## 課題

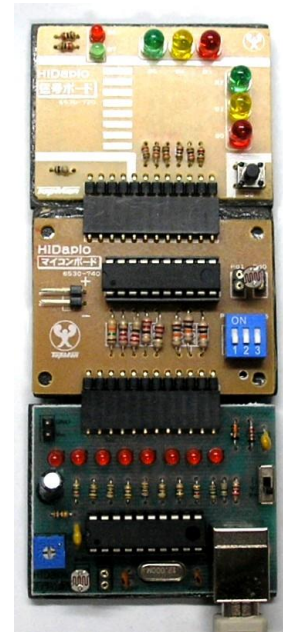
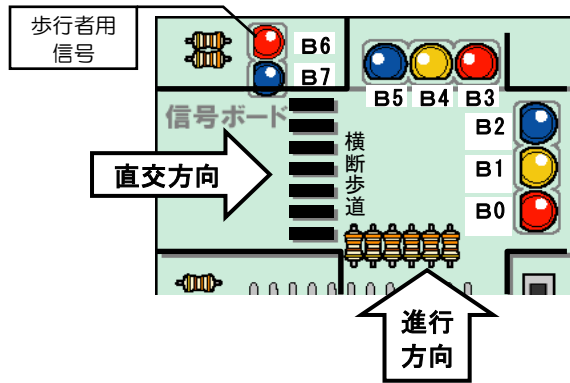
電子さいころのプログラムを考えてみよう。



結果と感想

さいころのように  
明るくしたところで  
数字が出る

## 2. 信号機

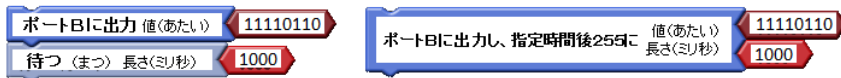


※信号機ボードは車用の信号と歩行者用の信号があります。  
 車用の信号は進行方向が B3～B5 ピンに、直交方向に B0～B2  
 ピン、歩行者用信号が B6～B7 ピンにつながっています。

### プログラムの作成から動作まで

「ArduBlock」でプログラムを作成する。

(例) 進行方向「赤」 直行方向「赤」を1秒間点灯する



(解説) B0とB3を「0」＝「点灯」にする。(上の左右どちらのブロックでもよい)

### 課題

実際の信号機のように、点滅時間も考えてプログラムしよう

(1) 最初は時間を一定にして、考えてみましょう。

動作										ポートBの値						
B5	B4	B3	B2	B1	B0	進行 方向	直交 方向	時間								
青	黄	赤	青	黄	赤											
○	○	●	○	○	●	赤	赤	1000	1	1	1	0	1	1	0	
○	○	●	●	○	○		青	1000	1	1	1	0	0	1	1	
○	○	●	○	●	○		黄	1000	1	1	1	0	1	0	1	
○	○	●	○	○	●	赤	赤	1000	1	1	1	0	1	1	0	
●	○	○	○	○	○		青	1000	1	1	0	1	1	1	1	0
○	●	○	○	○	○		黄	1000	1	1	1	0	1	1	1	0



(2) 続いて実際の信号のように時間を変えてみましょう。

動作										ポートBの値		
B5	B4	B3	B2	B1	B0	進行方向	直交方向	時間				
青	黄	赤	青	黄	赤							
						赤	赤	1000	1 1 1 1 0 1 1 0			
							青	6000	1 1 1 1 0 0 1 1			
							黄	2000	1 1 1 1 0 1 0 1			
						赤	赤	1000	1 1 1 1 0 1 1 0			
								青	6000	1 1 0 1 1 1 1 0		
								黄	2000	1 1 1 0 1 1 1 0		

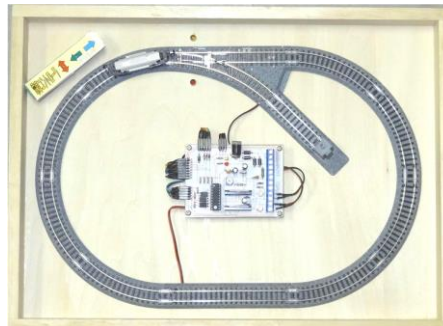
(3) 歩行者用信号も使った点滅も考えてみましょう。

動作										ポートBの値				
B7	B6	B5	B4	B3	B2	B1	B0	歩行者	進行方向	直交方向	時間			
青	赤	青	黄	赤	青	黄	赤							
								赤	赤	赤	1000	1 0 1 1 0 1 1 0		
											青	6000	1 0 1 1 0 0 1 1	
											黄	2000	1 0 1 1 0 1 0 1	
											赤	1000	1 0 1 1 0 1 1 0	
								青	青	青	3500	0 1 0 1 1 1 1 0		
											×	500	1 1 0 1 1 1 1 0	
											青	500	0 1 0 1 1 1 1 0	
											×	500	1 1 0 1 1 1 1 0	
											青	500	0 1 0 1 1 1 1 0	
											×	500	1 1 0 1 1 1 1 0	
											赤	黄	2000	1 0 1 0 1 1 1 0

### 3. N ゲージ

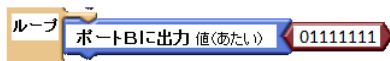
※ 線路への電力供給が B7 と B6 ピン、ポイントの切り替えが B5 ピンにつながっています。

	B7 (High,1)	B7 (Low,0)
B6 (High,1)	停止	右回り
B6 (Low,0)	左回り	停止



#### プログラムの作成から動作まで

①「ArduBlock」で電車を走らせるプログラムを作成する。



②B5 ピンを Low (0) にすると、ポイントが切り換わります。

③電車の走行中、雑音が発生して暴走することがあります。その時は、電源を入れなおしましょう。

#### 課題 1

下のプログラムを入れて、電車をゆっくり走らせてみよう。



※、電流を頻繁に ON-OFF する制御を PWM 制御といい、扇風機の回転速度の変換や、照明器具の明るさ調整などに使われています。

#### 課題 2

反対方向に走らせたり、ポイントに引き込んだりしてみよう。

結果

※N ゲージ制御装置は、テクノキット社製です。

## 4. 電子オルゴール

### 課題

いろいろな曲を入力して、鳴らしてみよう。またセンサを使って、明るくなると演奏するなど、試してみよう。

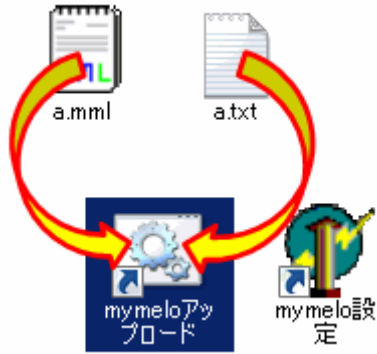
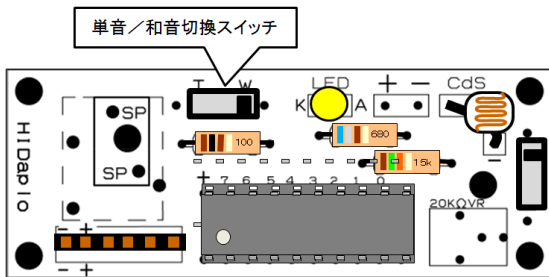
### 和音の電子オルゴール

和音／単音切り替えスイッチを和音側（右側）にして、右のようなテキストファイルをドラッグ&ドロップすることで、和音の鳴るオルゴールにすることができます。

※詳細は「ヒダピオ」の Web ページを参照。

```

トラック 1
ふあ2ふあら しらしん
トラック 2
ん2んん みふあみん
トラック 3
↓ふあ2ふあ2 みふあみん
    
```



### 単音の電子オルゴール

「ヒダピオ」の「電子オルゴール」を使うと、

- ・簡単にデータの入力
- ・プログラムの作成、コンパイル、保存

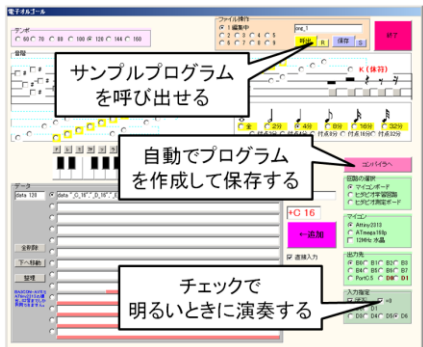
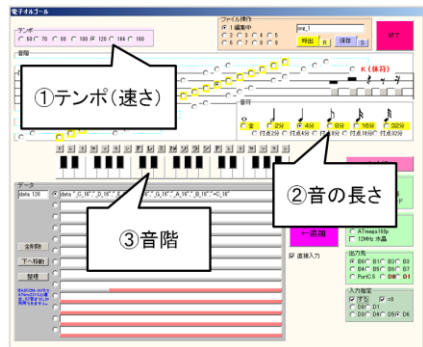
ができます。



①自分の作りたいオルゴール用の音楽のテンポデータ、音符データを入力する。

※ 回路の選択を「マイコンボード」にする。

②プログラムの保存、コンパイル、書き込み、動作の確認



# 参考資料

## 1. 2進数

### 10進数と2進数

※10進数と10のべき乗

(例) 10進数の427は	$427 = 400 + 20 + 7$ $= 4 \times 100 + 2 \times 10 + 7 \times 1$ $= 4 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$
---------------	--

※2進数と2のべき乗

(例) 2進数の101は	$101 = 1 + 0 + 1$ $= 1 \times 4 + 0 \times 2 + 1 \times 1$ $= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
--------------	---

### 2進数から10進数への変換

<p>(例) 2進数の110は</p> $1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ $= 1 \times 4 + 1 \times 2 + 0 \times 1$ $= 4 + 2 + 0$ $= 6$ <div style="border: 2px solid blue; padding: 5px; display: inline-block;"><math>110_{(2)} = 6_{(10)}</math></div>	<p>(例) 2進数の1010は</p> $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$ $= 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$ $= 8 + 0 + 2 + 0$ $= (10)$ <div style="border: 2px solid blue; padding: 5px; display: inline-block;"><math>1010_{(2)} = (10)_{(10)}</math></div>
---	---

### 10進数から2進数への変換

<p>10進数の427は</p> $\begin{array}{r} 10 \overline{) 427} \\ 10 \overline{) 42} \quad \dots 7 \\ 10 \overline{) 4} \quad \dots 2 \\ 0 \quad \dots 4 \end{array}$ <div style="text-align: right; color: green;">↑</div>	<p>(例) 10進数の6は</p> $\begin{array}{r} 2 \overline{) 6} \\ 2 \overline{) 3} \quad \dots 0 \\ 2 \overline{) 1} \quad \dots 1 \\ 0 \quad \dots 1 \end{array}$ <div style="text-align: right; color: green;">↑</div> <div style="border: 2px solid blue; padding: 5px; display: inline-block;"><math>6_{(10)} = (110)_{(2)}</math></div>	<p>(例) 10進数の10は</p> $\begin{array}{r} 2 \overline{) 10} \\ 2 \overline{) 5} \quad \dots 0 \\ 2 \overline{) 2} \quad \dots 1 \\ 2 \overline{) 1} \quad \dots 0 \\ 0 \quad \dots 1 \end{array}$ <div style="text-align: right; color: green;">↑</div> <div style="border: 2px solid blue; padding: 5px; display: inline-block;"><math>10_{(10)} = (1010)_{(2)}</math></div>
---	---	---

※10進数を2で割っていき、余りを下から並べると2進数になります。

## 2. 2進数とコンピュータ

現在多く使われているコンピュータは、

- 電圧が高い（たとえば+5V）
- 電圧が低い（0V）



1010101110111  
0011101001010  
111110111101  
0101110111001

の二つの状態の組み合わせで、いろいろな状態を表し、計算をしています。そのため、0と1の組み合わせで数値を表す2進数と相性がよいことが知られています。10進数の1桁には0から9の数値が入りますが、2進数の1桁には0か1が入ります。この1桁分の情報をビットという単位で数えます。

## 3. 表計算ソフトでのプログラム例

プログラムは色々なところで使われます。

下の図は、表計算ソフトでプログラムを使って入力した例です。

	A	B	C	D	E	F	G
1		2					
2		4					
3		6	CommandButton1				
4		8					
5		10					
6		12					
7		14					
8		16					
9		18					
10		20					
11		22					

```
Private Sub CommandButton1_Click()  
    For i = 1 To 200  
        Cells(i, 1) = i * 2  
    Next i  
End Sub
```

ここでは、For~Next文を使って、ボタンを押すと、一瞬にして「行番号の2倍の数を入力」しています。

またIf文を使うと、「3の倍数の行には、行番号の2倍の数値を、それ以外には行番号の数を入力」するような、複雑なこともできます。

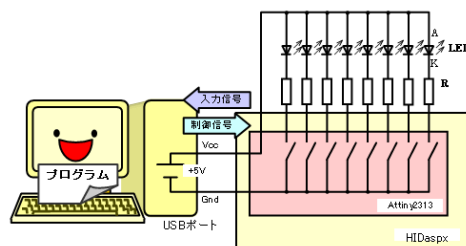
プログラムを知っておくと、コンピュータをより便利に利用できます。

## 4. HIDaspX

HIDaspX（エイチ・アイ・ディーアスペックス）とはAVRマイコンを利用したHIDクラス対応の回路で、AVRマイコンにプログラムを書き込むライターだけでなく、USBポートを利用したパソコンによる制御（USB-I/O）を可能にすることができます。



HIDaspX を使った USB-I/O

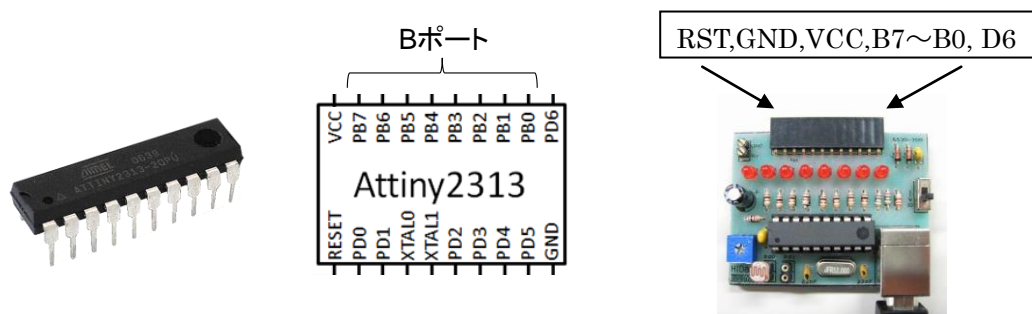


## 5. ATtiny2313 とマイコンボード

マイコンには、汎用または専用のものが、国内外各社で各種生産されています。

制御学習では、操作性や価格、入手のしやすさから、初心者が利用しやすいものとして PIC (ピック) マイコンや AVR (エーブイアール) マイコンなどが利用されます。

ヒダピオシステムでは、AVR マイコンの中からピン配置が分かりやすく、安価で入手しやすい ATtiny2313 (エーティータイニー・ニーサンイチサン) または ATtiny4313 を利用します。 ※ATtiny4313は2313と同等の働きをし、2313の2倍のメモリがあります。



ヒダピオシステムで利用するマイコンボードは、ヒダピオ学習回路を使って簡単に書き込みして、7 セグメントや LED 信号機ボード、ディスプレイボードなどを動作させることができます。

## 謝辞

ヒダピオシステムで使用する HIDaspx および周辺ツール一式は、山形県立産業技術短期大学校情報制御システム科千秋広幸教授が中心となって、多くの方の手によって開発されました。(開発者および経緯の詳細は hidspax 附属の説明書をご覧ください)

中学生向けの教材として、各種回路や「HAG\_mini」については元大阪府立砂川高等学校長 井本泰彦氏が、ヒダピオ専用コンパイラ「バスコンピ」は北海道業能力開発大学校電子情報技術科 谷岡 政宏氏が開発していただきました。また、「ArduBlock」の改変は大阪教育大学 光永法明准教授、教材一式の評価については三田市技術科の先生方の他、トップマン、テクノキットなど多くの方のご援助ならびにご協力をいただきました。

これまで制御の学習をする上で、学習教材が高価であったり、取り扱いが面倒なことが障害となっていました。千秋教授をはじめとする多くの方々のお陰で、これほどまでに内容の深い学習が安価で手軽にできるようになりました。皆さま方のご研究に心から敬意を表するとともに、またご協力いただきましたことに篤くお礼を申し上げます。

JA制御ヒダピオシステム教材 コンピュータ制御を学ぼう

監修

浅田寿展(兵庫県三田市立中学校 技術科教諭)

【本書の無断転載を禁ず】

著作