総合理工学実験実習（情報システム系） 実験書

目次

IchigoJam は株式会社 jig.jp の登録商標です.

本実験書を許可なく, 一部または全部を転載・複製することを禁じます.

著作者: 萩原 涼介 – me@raryosu.info

1. 実験について

［実験の目的］

IchigoJam プリント基板キットを組み立て，ハンダ付けの方法を体験する。また，プログラムの基本となる BASIC 言語を用いて，プログラミングの実習を行い，プログラム作成の基本を学ぶ。

［IchigoJam について］

IchigoJam は手軽にプログラミングを体験することができるマイコンボードである。起動するとすぐに BASIC という言語でプログラムを書くことができ，動かすことができる。

［注意事項］

IchigoJam は，精密な電子回路であり，高い熱や静電気，振動に強くない。ハンダ付け実習において，［別紙］はんだ付けのしおりをよく読みハンダ付けを行うこと。

［持参品］

実験時には必ず以下の物を持参すること。

- 実験書
- IchigoJam プリント基板キット
- MicroUSB ケーブル（MicroUSB と USB の端子がついたケーブル）
- シリアル通信ケーブル（黒いケーブル）

［参考文献］

[1] jig.jp, "こどもパソコン IchigoJam", http://ichigojam.net/, 2016.

[2]蘆田昇, "IchigoJam でプログラミング", プログラミングクラブ ネットワーク, 2015.

2. 小型パソコン IchigoJam の製作

2.1 実験器具

IchigoJam プリント基板キット，はんだごて，はんだ，はんだごて台，ニッパ

2.2 部品の確認

袋の中に，次の図 2.1 に示されている部品がすべて揃っているか確認する。部品の名前は表 2.1 に示す。

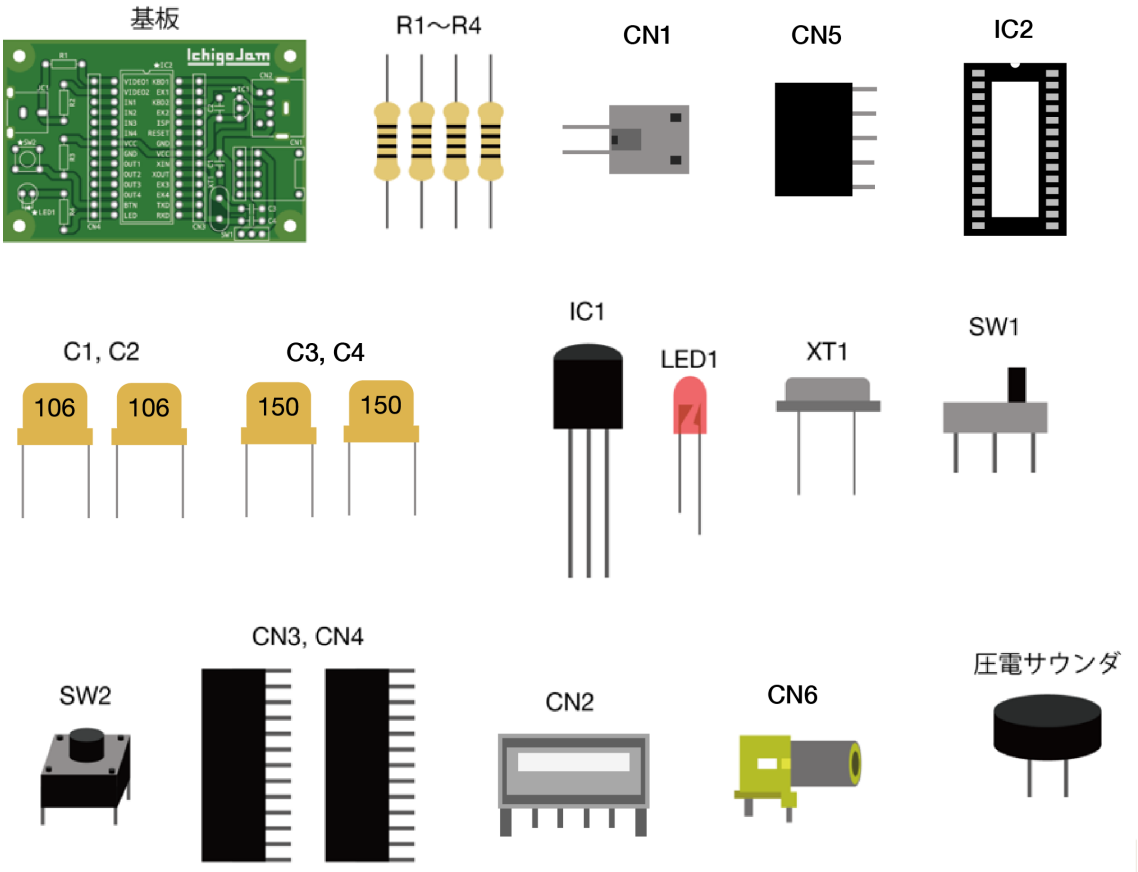

図 2.1 IchigoJam 組み立てキットの部品一覧

表 2.1 部品一覧と組み立て推奨順

順番	記号	名称	備考
1	R1	抵抗 470Ω	黄紫茶金
2	R2	抵抗 100Ω	茶黒茶金
3	R3	抵抗 1MΩ	茶黒緑金
4	R4	抵抗 330Ω	橙橙茶金
5	CN1	microUSB 端子	給電用
6	XT1	クリスタル 12MHz	
7	C1, C2	セラミックコンデンサ(106)	10μF 2 つ
8	C3, C4	セラミックコンデンサ(15 又は 150)	15pF 2 つ
9	IC2	IC ソケット	
10	SW2	タクトスイッチ(ボタン)	脚が左右に開くように取り付ける
11	CN2	USB 端子	キーボード用
12	IC1	三端子レギュレータ	向きに注意
13	CN3, CN4	ピンソケット(14 ピン)	2 つ
14	CN5	ピンソケット(5 ピン)	
15	LED1	赤色 LED	脚が長い方が内側
16	CN6	ビデオ端子	
17	SW1	スライドスイッチ	向き無し
		IchigoJam ステッカー	
		圧電サウンダー	3週目以降で利用する

参考 -SI 接頭辞-

物理量に対して10の何乗かを示すための接頭辞。

キロ(k) = 10^3	ミリ(m) = 10^{-3}
メガ(M) = 10^6	マイクロ(μ) = 10^{-6}
ギガ(G) = 10^9	ナノ(n) = 10^{-9}
テラ(T) = 10^{12}	ピコ(p) = 10^{-12}

抵抗の色の帯は，その抵抗の抵抗値と誤差を示している。次の表 2.2 を参考にして IchigoJam の組み立てに使う抵抗の値を読んでみよう。なお，抵抗のカラーコードが 茶黒赤金 のときの例を図 2.2 に示す。

表 2.2 抵抗のカラーコード

色	第1数字	第2数字	第3数字（乗数）	誤差(%)
黒	0	0	$10^0 = 1$	
茶	1	1	$10^1 = 10$	±1
赤	2	2	$10^2 = 100$	±2
橙	3	3	$10^3 = 1{,}000$	
黄	4	4	$10^4 = 10{,}000$	
緑	5	5	$10^5 = 100{,}000$	
青	6	6	$10^6 = 1{,}000{,}000$	
紫	7	7	$10^7 = 10{,}000{,}000$	
灰	8	8	$10^8 = 100{,}000{,}000$	
白	9	9	$10^9 = 1{,}000{,}000{,}000$	
金	-	-	$10^{-1} = 0.1$	±5
銀	-	-	$10^{-2} = 0.01$	±10
無	-	-	-	±20

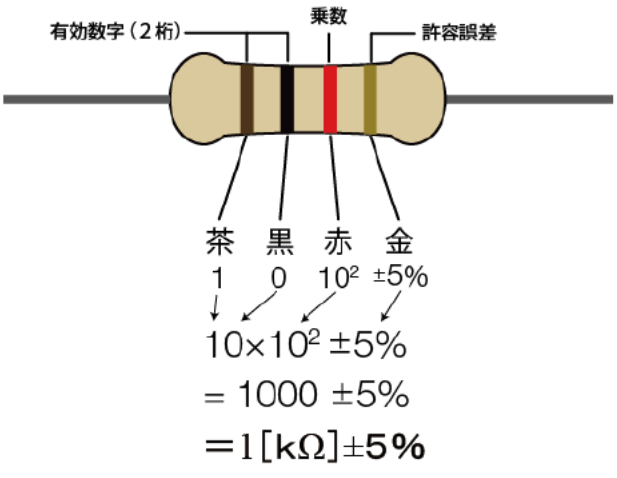

図 2.2 抵抗のカラーコードの例

2.3 IchigoJam の制作

これから先は，IchigoJam の基板にそれぞれの部品をはんだ付けし，実際に動作するか確認していく。

以下に示す手順ではんだ付けを行う。必ず**良く読んで**はんだ付けすること。

手順 1. 抵抗をはんだ付けする

R1, R2, R3, R4 に抵抗をはんだづけする。位置を間違えないよう注意すること。向きはないため，好きな向きではんだ付けする。

手順 2. MicroUSB 端子(給電用)をはんだ付けする

後ろの脚を基板にさし込み，垂直に曲げて裏からはんだ付けする。取れやすいため，多めにはんだを流し込む。

手順 3. クリスタルをはんだ付けする

クリスタルを XT1 に取り付けてはんだ付けする。

手順 4. セラミックコンデンサ(106)をはんだ付けする

10μF のセラミックコンデンサ(106)を C1, C2 にはんだ付けする。誤って 150(15)と書かれたセラミックコンデンサをはんだ付けしないよう気をつける。また，無理に深くまで押し込まずにはんだ付けする。

手順 5. セラミックコンデンサ(150(15))をはんだ付けする

15pF のセラミックコンデンサ(150(15))を C3, C4 にはんだ付けする。

手順 6. IC ソケットを取り付ける

切り欠きが上に来るようにはんだ付けする。4隅からはんだ付け(右上→左下→右下→左上)するようにすると良い。

手順 7. タクトスイッチをはんだ付けする

SW2 にタクトスイッチ(ボタン)をはんだ付けする。脚が開く方向が左右になるようにする。

手順 8. USB コネクタ(キーボード用)を取り付ける

CN2 にはんだ付けする。はんだを多めに流し込む。

手順 9. 三端子レギュレータを取り付ける

IC1 にはんだ付けする。**平たいほうが内側に来るように**はんだ付けする。

手順 10. ピンソケットを取り付ける

CN3, 4 に 14 ピン，CN5 に 5 ピンのピンソケットをはんだ付けする。

手順 11. 赤色 LED を取り付ける

LED1 に赤色 LED を取り付ける。**脚が長いほうが内側に来るように**する。

手順 12. ビデオ端子をはんだ付けする

CN6 にはんだ付けする。はんだを多めに流し込む。

手順 13. スライドスイッチを取り付ける

SW1 に以下の手順ではんだ付けする。

(a) 真ん中のピンに少量のはんだ付けをする。

(b) 傾いていないか確認し，傾いていれば，真ん中を温め直してまっすぐにする。

(c) まっすぐになったら両端のはんだ付けを行う。

傾いていると，スイッチの故障につながるため，なるべくまっすぐつけるようにする。

2.4 動作確認

RCA ケーブル（黄色）をモニタと IchigoJam 基板のそれぞれのビデオ端子に，電源に接続した MicroUSB ケーブルを MicroUSB 端子に接続する。さらに、USB 端子にキーボードを接続する。

その後，スライドスイッチを右側にスライドするとモニタが図 2.3 のような画面になることを確認する。確認できたら，リスト 2.1 に示すコマンドを入力し，LED がきちんと点灯・消灯することを確認する。

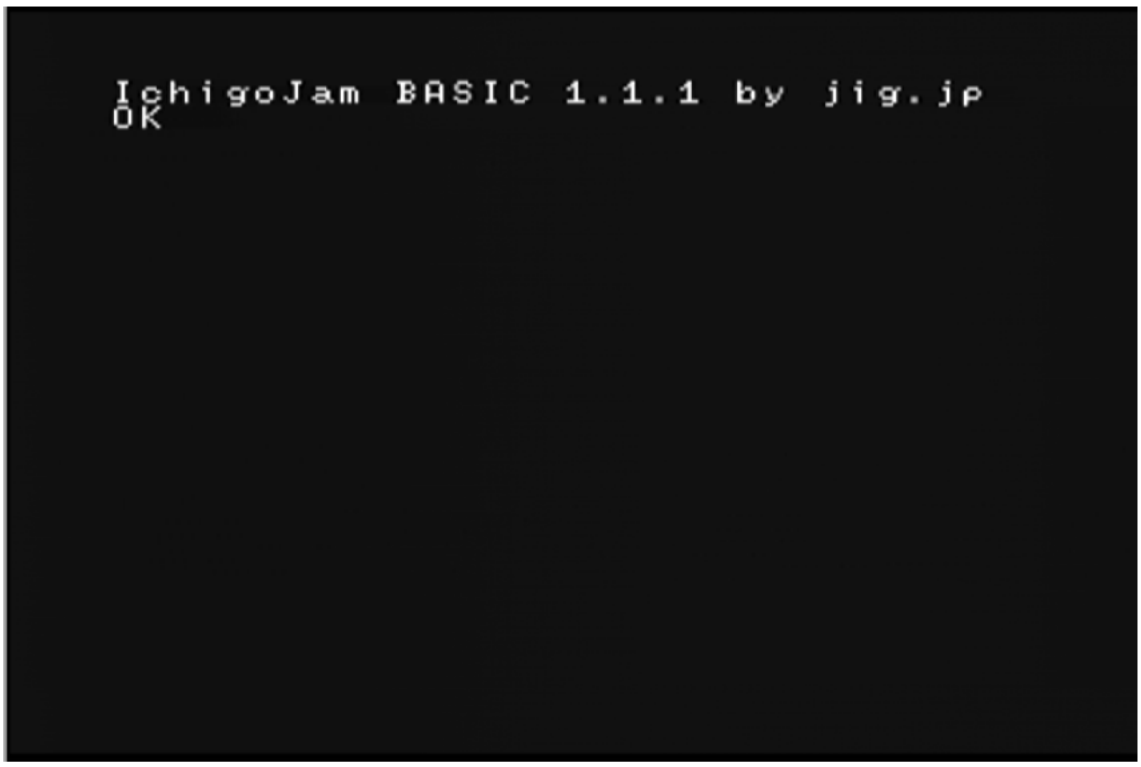

図 2.3 IchigoJam を起動した時のモニタの様子

リスト 2.1 LED を点灯・消灯させるプログラム

```
LED 1  (LED が点灯する)
LED 0  (LED が消灯する)
```

3. BASIC プログラミング演習（基礎）

3.1 BASIC とは

BASIC は **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode（初心者向け汎用記号命令コード）の略称で，その名の通り初心者でも簡単にプログラムを書くことができるプログラミング言語である。

これからの 4 週は，先週までに作成した IchigoJam 上で BASIC のプログラムを実際に動作させながらプログラミングの基礎を学んでいく。

3.2 BASIC のプログラムの書き方

BASIC のプログラムの例をリスト 3.1 に示す。

リスト 3.1 LED を点滅させるプログラム

```
10 LED 0
20 WAIT 30
30 LED 1
40 WAIT 30
50 GOTO 10
60 END
```

BASIC のプログラムは　行番号 命令の種類 命令に必要な値　という行の組み合わせで成り立っている。プログラムは行番号の小さいものから順に実行される。

命令の種類を　関数　とよび，命令に必要な値は　引数（ひきすう）　とよぶ。

IchigoJam で使うことのできる関数一覧は，IchigoJam 組み立てキットに同封されている説明書の裏面に記載されている。演習中わからない関数が出てきた時には，説明書を確認すること。

3.3 PC との接続

シリアル通信ケーブルを用いて PC に IchigoJam を接続する。まず，シリアル通信ケーブルとジャンパーコードを用意し，表 3.1 のように接続する。さらに、シリアル通信ケーブルの USB コネクタは PC の USB ポートへ接続する

表 3.1 シリアル通信ケーブルと IchigoJam の接続

シリアル通信ケーブル	−	IchigoJam
TXD（オレンジ）	−	RXD
RXD（黄色）	−	TXD
GND（黒色）	−	GND

次に，MicroUSB ケーブルを IchigoJam の MicroUSB 端子と PC の USB ポートにそれぞれ接続する。そして，IJUtilities*を起動し，ポートを選択し，IchigoJam との通信を開始する。ポートは，**COM1 以外のもの(COM3 など)を選択**する。

IchigoJam のスライドスイッチを左にスライドすると，図 3.1 のような画面になることを確認する。

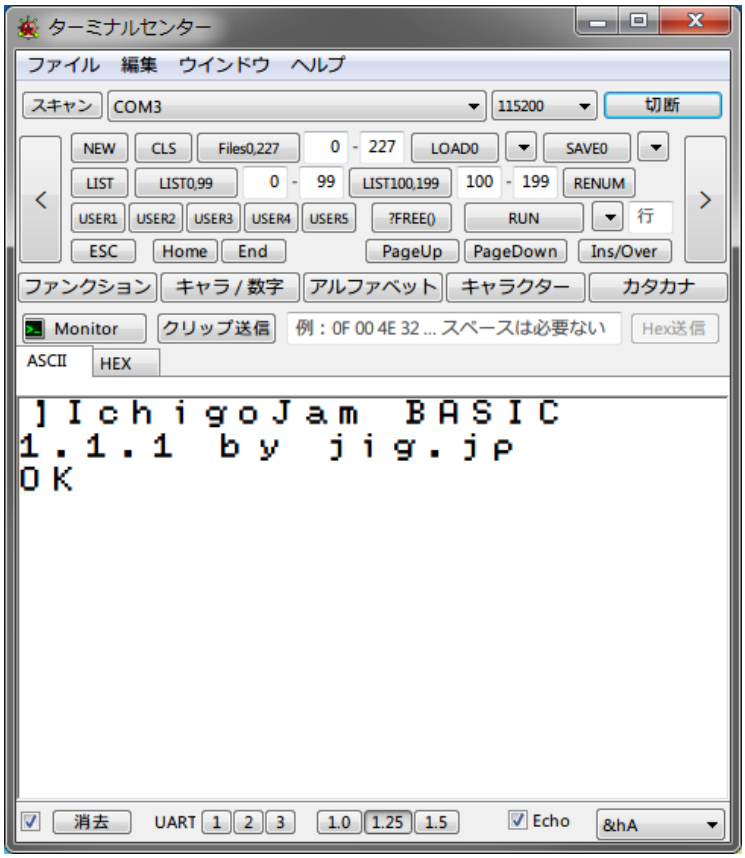

図 3.1 IchigoJam を起動した時の IJUtilities の表示

* IJUtilities は Micono Utilities さんが作成しているツールです.
http://ijutilities.micutil.com/

3.4 IJUtilities の使い方

①IJUtilities を起動し，IchigoJam の電源を入れる

② ツール ウィンドウ(細い画面)の 新規 ボタンを押す

③表示されたウィンドウ(図 3.2)にプログラムを打つ

④打ち終わったら すべて送信 ボタンを押し， RUN ボタンを押す(図 3-1)

⑤正しく実行できたら，プログラムを入力したウィンドウで，プログラムをコピーし，Word に貼り付ける(プログラムを入力したウィンドウで右クリック→ すべてを選択 → コピー ，Word の画面で右クリック→ 貼り付け)

⑥もし，正しく実行できず，プログラムが終了されなくなった場合，ESC キーを押してプログラムの実行を終了させる。

⑦**NEW ボタンを押して**，次のプログラムを作成する。

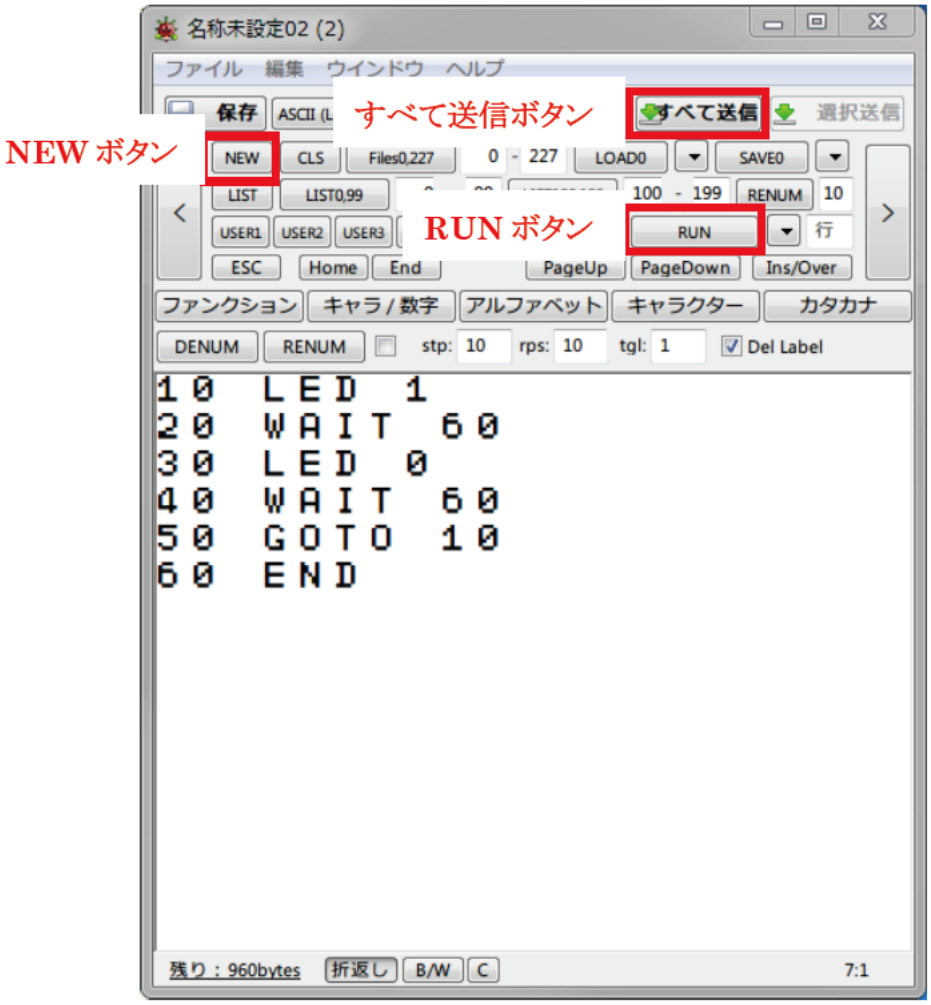

図 3.2 プログラム入力ウィンドウ

以降は BASIC を用いたプログラムの演習を示す。演習ごとに演習番号とソースコードを Word に貼り付け，図 3.3 の書式で報告せよ。なお，提出先やファイル名は以下のようにすること。

提出先（例）:

ReadWrite(des16)(Y:)¥総合理工学科¥情報システム系¥1 年実験実習¥T1¥1-1-40 名前

ファイル名(例):

課題 1.docx

1 年○組○番 名前

PG1-1

10 LED 0

20 WAIT 30

30 LED 1

40 WAIT 40

50 GOTO 10

60 END

[結果] LED が点滅した

PG1-2

・・・・

図 3.3 報告書の書式

PG1-1. LED の点滅

リスト 3.2 に示すプログラムを入力し，LED が点滅することを確認しなさい。

リスト 3.2 LED を点滅させるプログラム

```
10 LED 0
20 WAIT 60
30 LED 1
40 WAIT 60
50 GOTO 10
60 END
```

PG1-2. 文字の表示 1

リスト 3.3 に示すプログラムを入力し，画面上に文字が表示されることを確認しなさい。

リスト 3.3 文字を表示するプログラム

```
10 PRINT "Hello World!"
20 END
```

PG1-3. 文字の表示 2

リスト 3.3 のプログラムを参考に，自分の名前（アルファベット）を画面に表示させなさい。

PG1-4. 模様の表示

リスト 3.4 に示すプログラムを入力し，画面上に模様が表示されることを確認しなさい。

リスト 3.4 模様を表示するプログラム

```
10 PRINT "***"
20 PRINT "==="
30 PRINT "+++"
40 END
```

PG1-5. 四則演算 1

リスト 3.5 に示すプログラムを入力し，計算結果が正しいことを確認しなさい。**A%B は A を B で割った値のあまりを表す。**

リスト 3.5 四則演算をするプログラム

```
10 PRINT 5+3
20 PRINT 10-2
30 PRINT 2*4
40 PRINT 32/4
50 PRINT 17%9
60 END
```

PG1-6(応用). 四則演算 2

リスト 3.5 の計算式の数字を変更して, 結果が正しいことを確認しなさい。なお, IchigoJam の BASIC では計算に整数しか用いることができず, 割り算の結果が小数になる場合は**小数点以下が切り捨てられる。**

3.5 変数

数学の授業では，数式の中で数字の代わりに x や y といった文字を用いてきた。プログラムにおいても， 変数 というものが用意されている。変数も数式中の文字と同じように数字を入れておく箱として使うことができる。リスト 3.6 に示すプログラムは，変数を用いて長方形の面積を計算するプログラムである。

リスト 3.6 長方形の面積を計算するプログラム

```
10 H=4
20 W=5
30 A=H*W
40 PRINT A
50 END
```

このプログラムでは，H（HEIGHT; 高さ）という変数に 4 を，W（WIDTH; 幅）に 5 を 代入 し，A（AREA; 面積）という変数にそれらを掛けたものを代入している。 代入 とは変数という箱に数字を入れることである。変数をプログラムに用いることで，プログラムが何をしているかわかりやすくし，プログラム中の値を変更しやすくするという効果もある。IchigoJam の BASIC では，**変数名にはアルファベット1文字のみを用いる**ことができ，変数には-32767 から 32767 までの整数を代入することができる。

PG2-1. 変数を用いた四則演算 1

次に示すプログラムを入力し，計算結果が正しいかどうかを確認しなさい。

リスト 3.7 変数を用いた足し算

```
10 X=10
20 Y=5
30 Z=X+Y
40 PRINT Z
50 END
```

PG2-2. 変数を用いた四則演算 2

次のプログラムを実行し，計算結果を確認しなさい。

リスト 3.8 3変数の演算

```
10 X=5
20 Y=4
30 Z=3
40 PRINT X+Y*Z
50 PRINT (X+Y)*Z
60 END
```

PG2-3(応用). 円の面積と円周を求めるプログラム

半径が 6 の円の面積と円周を求めるプログラムを作成し，結果が正しいかを確認しなさい。なお，円周率を示す変数 P=3 とし，以下の形式で出力するプログラムにすること。

```
AREA: (面積)
LENGTH: (円周の長さ)
```

PRINT "AREA: "; (変数名)という形で文字列と変数名を;でつなぐことで，上記の出力ができる。

"(ダブルクオーテーション)"で囲まれた文字列は 文字 として認識され，そのまま表示される。逆に""で囲まない文字列は 変数 として認識される。

参考 -変数の名前の付け方-

変数に名前をつけるとき，a, b, c などのように意味のない文字を使うことがよくある。

しかし，プログラムを書くときにはなるべく意味のある文字列を使うほうが良いとされている。プログラムを何人かで書くときや，自分が昔に書いたプログラムを修正するときに，a, b, c だとわかりにくい。そのため，上の例では W や H などその変数が示すものの英単語の最初の文字を変数名に用いている。多くのプログラミング言語では 1 文字だけでなく何文字も使える事が多いため，なるべくわかりやすい変数名をつけるようにしよう(IchigoJam ではメモリ容量の関係で1文字しか用いることができない)。

3.6 条件分岐・ループ

条件分岐とは もしボタンを押されたら音を鳴らす などのように，条件式が成り立つか成り立たないかで実行するプログラムを分岐する命令のことである。BASIC には IF 命令が用意されている。

IF 命令を使ったプログラムの例と，その処理の流れ（フローチャート）を次に示す。

リスト 3.9 条件分岐のプログラム例 1

```
10 C=0
20 LED 1
30 WAIT 30
40 LED 0
50 WAIT 30
60 C=C+1
70 IF C<=5 GOTO 20
80 END
```

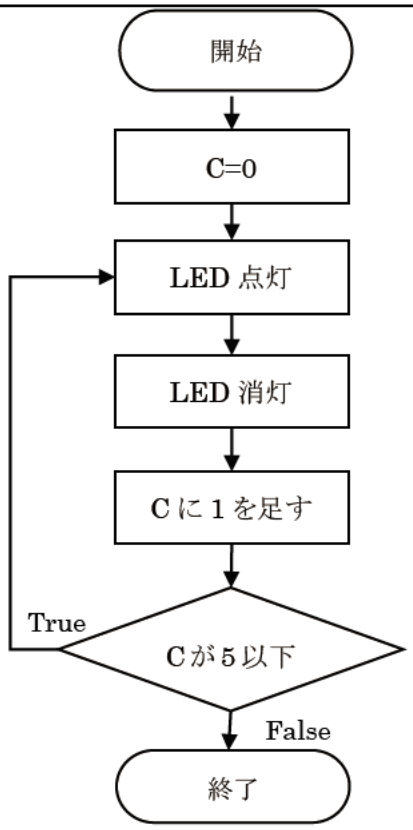

図 3.4 条件分岐のフローチャート

このプログラムはLEDの点灯・消灯をするたびに，カウント用の変数Cを1増やしていき，Cが5以下の間，その処理を繰り返す。数学では5以下を$C \leqq 5$と表現するが，BASICでは次の表のように条件を表現する。

表 3.2 条件式の記述ルール

条件式	意味	記述例
式1 == 式2	式1と式2の値が等しい	A==B
式1 != 式2	式1と式2の値が等しくない	A!=B
式1 <= 式2	式1の値が式2の値以下	A<=B
式1 >= 式2	式1の値が式2の値以上	A>=B
式1 < 式2	式1の値が式2の値より小さい	A<B
式1 > 式2	式1の値が式2の値より大きい	A>B

また，条件式が成り立つ時に複数の式を実行したい時には ： で命令をつなぐ。

PG3-1. 数値の判定プログラム

次に示すプログラムを入力し，実行しなさい。

リスト 3.11 数値判定プログラム

```
10 C=1
20 IF C==1 PRINT "#"
30 IF C==2 PRINT "*"
40 IF C==3 PRINT "="
50 WAIT 30
60 C=C+1
70 IF C<=3 GOTO 20
80 END
```

PG3-2. ボタンが押されたら LED を光らせるプログラム(1)

次に示すプログラムを入力し，本体のタクトスイッチを押した時に LED が点灯するかを確認しなさい。BTN() は，タクトスイッチが押されているなら1，そうでなければ 0 を表す。

リスト 3.11 ボタンが押されたら LED を光らせるプログラム

```
10 A=BTN()
20 IF A==1 LED 1
30 IF A==0 LED 0
40 GOTO 10
50 END
```

PG3-3. ボタンが押されたら LED を光らせるプログラム(2)

以下のプログラムは，リスト 3.11 と同様の動作をする。

```
10 LED BTN()
20 GOTO 10
30 END
```

3.7 圧電サウンダー

IchigoJam は外部に機器を接続し，制御することも可能である。ここでは，IchigoJam 組み立てキットに付属していた 圧電サウンダー を用いて，音を鳴らす。

圧電サウンダーの接続方法や操作方法は，付属している説明書を参照すること。

PG3-4. 音を鳴らそう 1

圧電サウンダーを IchigoJam の GND と SOUND にさして，次のプログラムを実行し，音がなることを確認しなさい。

リスト 3.12 音を鳴らすプログラム 1

```
10 PLAY "CDEFGABC"
20 END
```

PG3-5. 音を鳴らそう 2

次のプログラムを実行し，音がなることを確認しなさい。

リスト 3.13 音を鳴らすプログラム 2

```
10 PLAY "CDE3RCDE3RG4E4D4C4D4E4D4"
20 END
```

PG3-6(応用). 音を鳴らそう 3

次に示す条件を満たすプログラムを作成し，報告しなさい。

1. シラソファミレド という音を鳴らす
2. 1.を 5 回繰り返す
3. 必ず IF 文を用いる

参考 -IF 文とループ処理-

IF 文は もし○○ならば△△する という命令である。BASIC をはじめとする多くの言語には もし○○ならば△△する。**そうでなければ□□する。** ということを示すための IF ELSE 文が用意されている(BASIC では IF THEN ELSE 文)。例えば 奇数ならば#を表示する。そうでなければ*を表示する。 といったプログラムなら次のようになる。

```
10 C=1
20 IF C%2!=0 THEN PRINT "#" ELSE PRINT "*"
30 C=C+1
40 IF C<10 GOTO 20
50 END
```

難しそうに見えるかもしれないが，C を 2 で割ったあまりが 1 である数は奇数であり，その時(THEN)には # を表示し，そうでなければ(ELSE) * を表示している，ということがプログラムを読むことでわかる。

奇数を C%2==1(C%2!=1)や，2n+1 で表す手法はよく使うので，覚えておくと良い。

4. BASIC プログラミング演習（応用）

4.1 入力された数値の受け取り

プログラムを作成するとき，ユーザから数字や文字を受け取ることが多い。BASIC でも数字の受け取りが可能である。

リスト 4.1 入力された数値を受け取るプログラム

```
10 INPUT "ANS? ", A
20 PRINT A
30 END
```

リスト4.1を実行すると，**ANS?** と表示され，数値を入力すると，その数値が**A**に代入され，画面に表示される。**INPUT()**命令は数値のみに用いることができる。

演習 PG4-1. 数値を入力して処理を行うプログラム 1

以下のプログラムを入力し，正しく実行されることを確認しなさい。

リスト 4.2 入力された数値に対して四則演算を行うプログラム

```
10 INPUT "Number? ", N
20 PRINT "N="; N
30 PRINT "N+2= "; N+2
40 PRINT "N-1= "; N-1
50 PRINT "N*5= "; N*5
60 PRINT "N/3= "; N/3
70 PRINT "N%2= "; N%2
80 END
```

4.2 復習と応用

この節では，後半 4 週間で学んだことを参考に，自分で課題を解いていく。前半2問は基礎問題，後半2問は応用問題である。

PG4-2. 偶数・奇数を判断するプログラム 1

以下の処理の流れで，偶数のときと奇数の時で表示を変えるプログラムを作成しなさい。偶数は 2 で割ったあまりが 0 である値，奇数は 2 で割ったあまりが 1 である値であることを利用する。

1. 変数 C に1を代入する
2. その変数が偶数か奇数かを判定する
3. 偶数なら$を，奇数なら%を表示する
4. C の値を1増やす
5. 2〜4 の処理を 20 回繰り返す

PG4-3. 数当てゲーム 1

以下の処理の流れで数当てゲームを作成しなさい。

1. 変数 A に，答えの値を入力する
2. 友達に数値を入力してもらう
3. 入力された数値と変数 A を比較する
4. 一致したら Congratulations! と表示し，不正解なら 2, 3 を繰り返す。

PG4-4(応用). 偶数・奇数を判断するプログラム 2

以下を参考に，奇数が表示されているときには LED を点灯させるプログラムを作成しなさい。処理を一時停止させるためには WAIT 関数を用い，`WAIT 60` (60ms 処理を停止する)のように使う。

1. 変数 C に 1 を代入する
2. C の値を表示する
3. C の値が偶数なら LED を消灯し，奇数なら LED を点灯させる
4. 1 秒待つ
5. C の値を 1 増やす
6. 1〜5 の処理を 20 回繰り返す

参考 -WAIT 関数-

コンピュータの処理速度は非常に早く，IchigoJam はクロック周波数が 48MHz である。すなわち，1 秒間に $48*10^6$ 回もの処理を行っているということである。この速度で LED を点滅させると，人間の目には見えないほどの速さで点滅してしまい，正しく動作したかがわからない。そこで WAIT 関数を用い，その処理がきちんと行われているのかを確認したり，必要に応じて処理を先延ばししたりする。

PG4-5(応用)．数当てゲーム 2

これまでに習ったことを用いて，PG4-3 で作成した数当てゲームを改良しなさい。

例）正解のときに LED を光らせる，不正解だった回数を記憶しておいて最後に表示する，
乱数を用いる(下記参考を参照)，入力された値が答えより大きいか小さいか表示する 等

参考 -乱数-

乱数とは，ランダムな値のことで，プログラミング中にしばしば用いられる。BASIC にも RND 関数があり，`RND(6)`(0～6 のランダムな数値を返す)などのように使う。

4.3 [発展] FOR 文

この節は，上記の課題がすべて終了した人のための発展課題である。

さて，リスト 3.9 では，条件が成立したとき GOTO 命令を用いてループを実現しているが，多くのプログラムでは FOR 文と呼ばれる構文を用いる。以下に FOR 文を用いたプログラムを示す。

リスト 4.5 FOR 文のプログラム例 1

```
10 FOR C=0 TO 5
20   LED 1
30   WAIT 30
40   LED 0
50   WAIT 30
60 NEXT
70 END
```

FOR C=0 TO 5 は C を 0 から 5 まで 1 ずつ増やしていく命令である。

上記のプログラムでは 20 行目から 50 行目まで命令を実行した後，60 行目の NEXT 文で 10 行目に戻り，C に 1 を加算する。そのような処理を実行し，C=5 になるとループを抜け，70 行目によりプログラムの処理が終了する。

FOR NEXT 文は更に細かくプログラムを修正することができ，NEXT 文の実行時に C に加算する値を変更することができる。要するに 2 刻みで値を加算したり，5 刻みで値を加算したりできる。リスト 4.6 にサンプルプログラムを示す。

リスト 4.6 FOR 文のプログラム例 2

```
10 FOR C=1 TO 10 STEP 2
20   PRINT C
30 NEXT
40 END
```

リスト 4.6 を実行すると，2 刻みで値が表示される。具体的には， 1, 3, 5, 7, 9 となる。

以下では，FOR NEXT 文を利用した課題に挑戦してみよう。

PG4-5(応用). FOR 文を使ってみよう 1

リスト 4.5 に示したサンプルプログラムを実行し，LED が 5 度点滅することを確認せよ。

PG4-6(応用). FOR 文を使ってみよう 2

リスト 4.6 に示したサンプルプログラムを実行し，表示される内容を記録せよ。

PG4-7(応用). 数値判定プログラム

PG-10 リスト 3.11 に示したプログラムを FOR 文で書き直し，実行せよ。

PG4-8(応用). 偶奇判定プログラム

PG-11 参考 -IF 文とループ処理- 中で示したプログラムを FOR 文で書き直し，実行せよ。

PG4-9(応用). 九九を計算させよう

FOR 文を 2 つ入れ子にし，九九を計算するプログラムを作成せよ。なお以下のような出力にすること。

```
1*1=1
1*2=2
（中略）
9*9=81
```

なお，リスト 4.7 を実行すると，00 から 99 までの数値を表示することができる。

リスト 4.7 FOR 文のプログラム例 3

```
10 FOR A=0 TO 9
20   FOR B=0 TO 9
30     PRINT A; B
40   NEXT
50 NEXT
60 END
```

2016/03/17 R. Hagihara 作成

2016/03/31 N. Yanuki 編集

2016/05/19 N. Yanuki 修正

2017/XX/XX R. Hagihara 修正