

IchigoJam^{いちごじゃむ}で プログラミング

MISSION CARD POSSIBLE!



ゲームプログラミング

HEART♥6



ルーレットゲームを作ろう!!

君だけの
小っちゃな
コンピュータ。



2023.2.19 Ver.2.07

- ・このテキストは、保護者等の適切な指導のもとでのご利用を考えて製作しています。
ご利用によるすべての事故や損失に関しては、当方は一切の責任を負いません。
- ・本資料はCCライセンスならびに以下の規定にしたがって、複製・改変・再配布することが可能です。
著作権は放棄していません。
- ・「IchigoJam」は株式会社 jig.jp の登録商標です。
- ・タイトル、写真などに含まれる「IchigoJam」の称呼は全て株式会社 jig.jp の商品を示しています。
- ・本資料はNPO法人NEXTDAYの協力のもとNPO法人小樽青少年の科学の芽を育てる会が作成しました。
- ・資料の作成にあたり以下の資料を参照しました。
＞親子でベーシック入門 IchigoJamではじめてのプログラミング（出版社：ジャムハウス）
＞IchigoJamでプログラミング（発売：プログラミングクラブネットワーク）
- ・原稿についてはICHIGOJAM開発者福野 泰介様のブログを参照させて頂いてます。

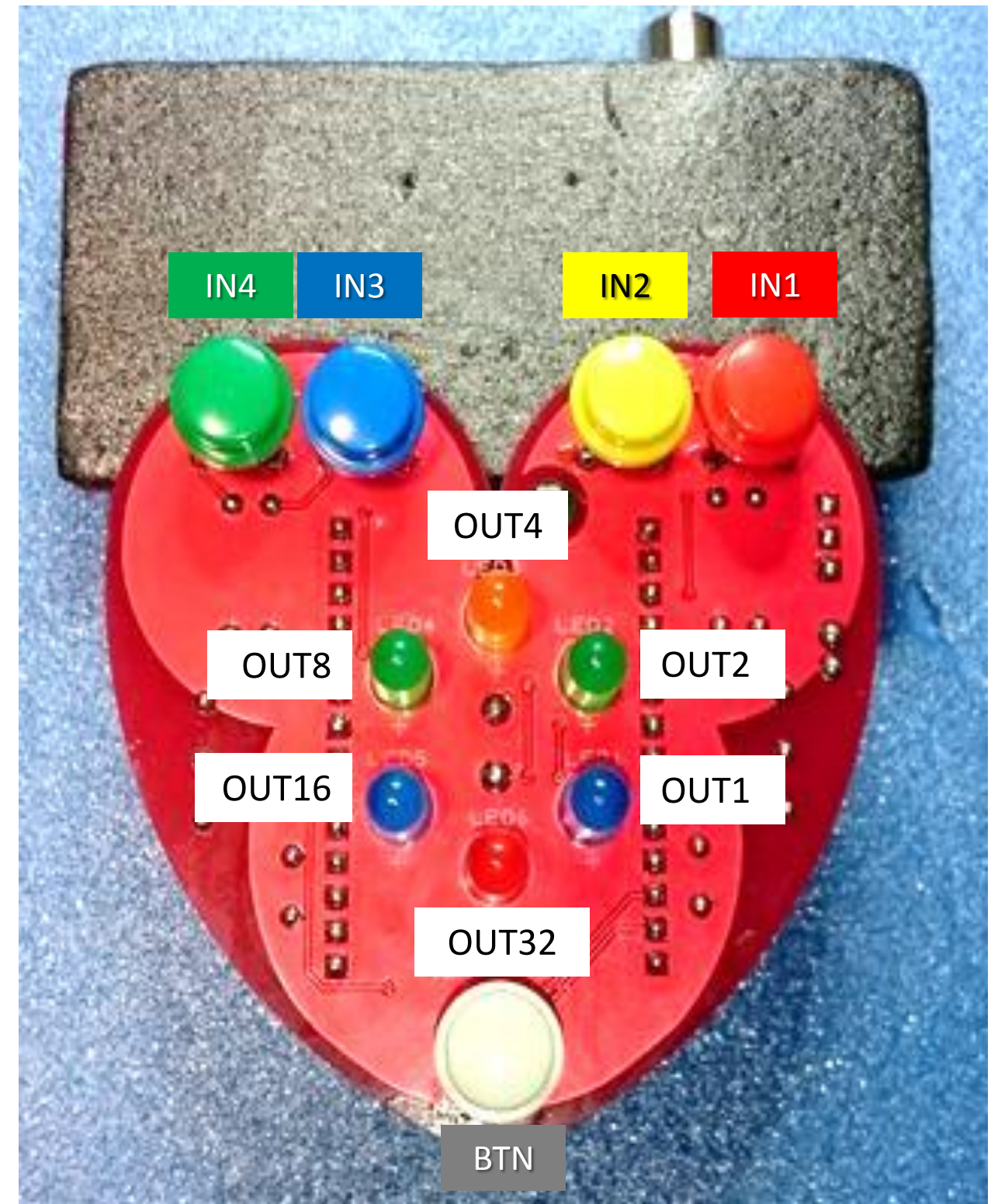
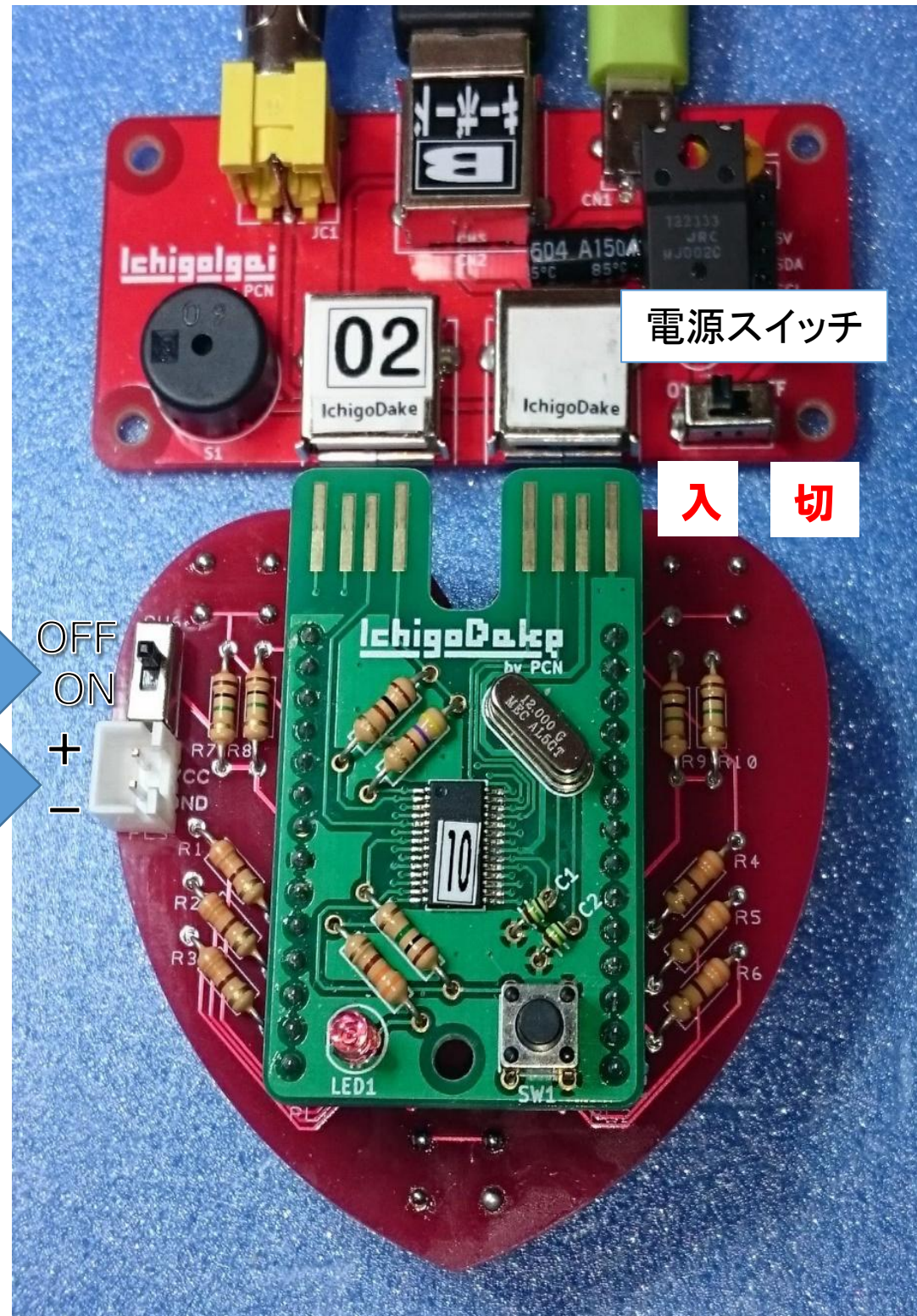
小樽別院 寺子屋教室

主催 小樽青少年の科学の芽を育てる会
協力 浄土真宗本願寺派 本願寺小樽別院
協力 NPO法人NEXTDAY

HEART♥6

モニター

電源



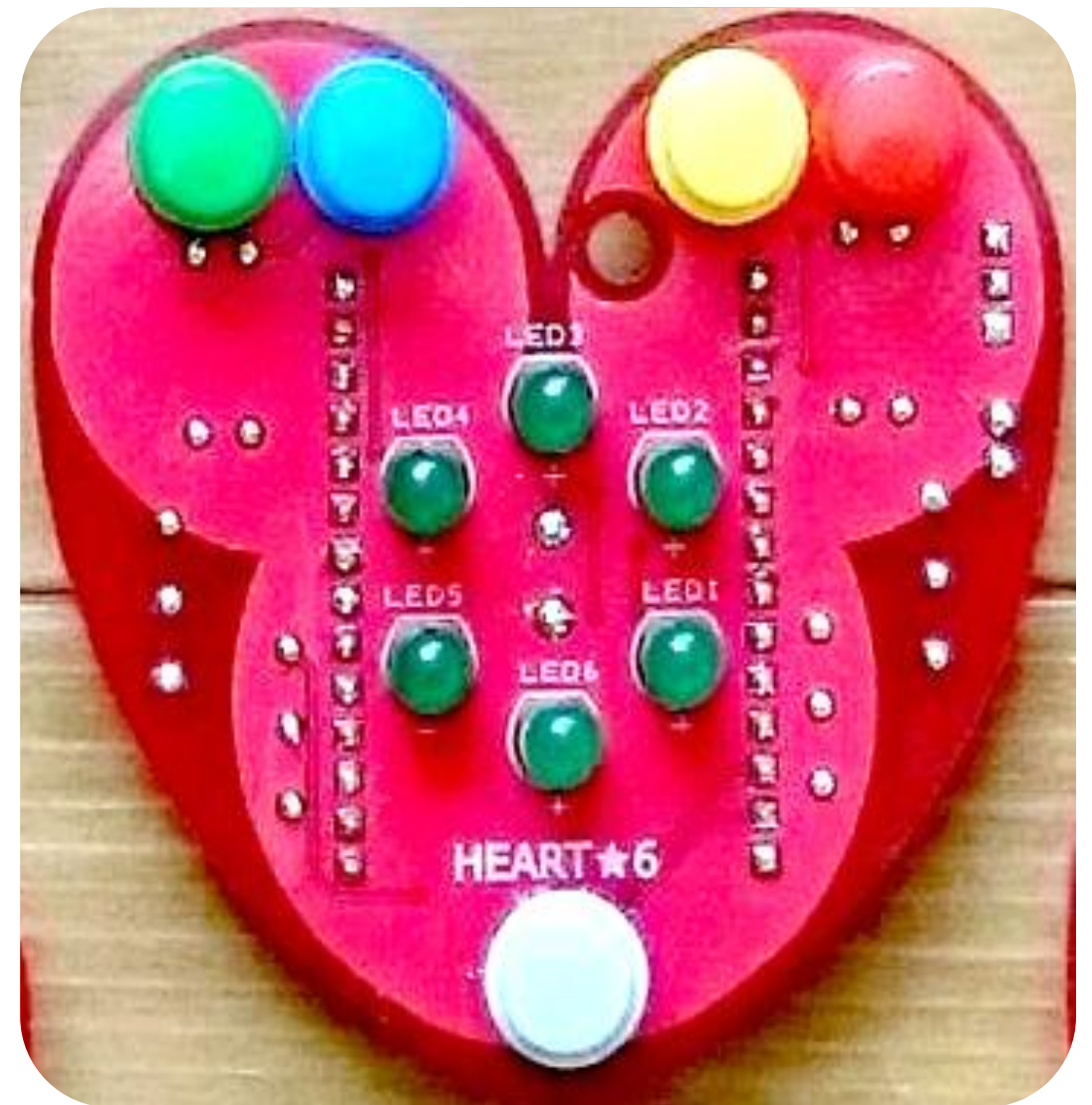
今日講座で使うICHIGOJAM基板とHEART基板です。

ウォーミングアップ

・LEDの点灯を調べよう

OUT	1	<input type="checkbox"/>
OUT	2	<input type="checkbox"/>
OUT	4	<input type="checkbox"/>
OUT	8	<input type="checkbox"/>
OUT	16	<input type="checkbox"/>
OUT	32	<input type="checkbox"/>

OUT命令でどのLEDが点灯するかメモしておきましょう



▪ 押しボタンを調べよう

```
10 ?IN(4),IN(3),IN(2),IN(1):WAIT5:GOTO 10
```

RUN



IN4

IN3

IN2

IN1

- 離す
- 押す

<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>



ボタンを押したときと離れたときの結果をメモしておきましょう

2進数と10進数で計算してみよう！

(2進数)

32	16	8	4	2	1
1	1	1	1	1	1

(10進数)

10000	1000	1000	100	10	1
1	1	1	1	1	1

2進数・10進数 手動変換

・2進数を10進数に変換

001010

2進数



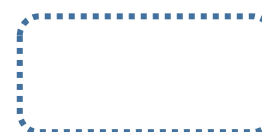
8+2=10

10進数

変換しよう

0 0 0 1 1 0

2進数



10進数

10進数	2進数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000



シフト演算

(シフト数)		(2進数)	(10進数)
$1 \ll \textcolor{red}{0}$:	000001 [●]	1
$1 \ll \textcolor{red}{1}$:	000010 [↷]	2
$1 \ll \textcolor{red}{2}$:	000100 ^{↷↷}	4
$1 \ll \textcolor{red}{3}$:	001000 ^{↷↷↷}	8
$1 \ll \textcolor{red}{4}$:	010000 ^{↷↷↷↷}	16
$1 \ll \textcolor{red}{5}$:	100000 ^{↷↷↷↷↷}	32

シフトすると桁位置が移動します



1. LEDを光らせる OUT 1<<N 命令を調べる

Nの値にて、どのLEDが点灯するか調べる

10 INPUT N

50 OUT 1<<N

F4

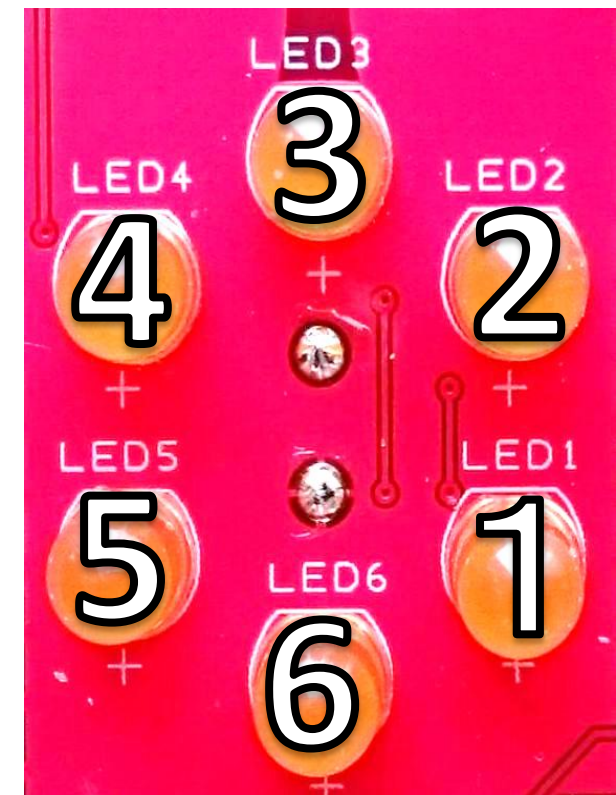
LIST

F5

RUN

LED (点いたLEDの○を●塗りつぶす)

N	6	5	4	3	2	1	N *
0	○	○	○	○	○	○	0+1 = 1
1	○	○	○	○	○	○	1+1 = 2
2	○	○	○	○	○	○	2+1 = 3
3	○	○	○	○	○	○	3+1 = 4
4	○	○	○	○	○	○	4+1 = 5
5	○	○	○	○	○	○	5+1 = 6
0	○	○	○	○	○	○	0+1 = 1 ?



6は点灯するLEDがないのでNの値を0にする

2. LEDをくるくる回すプログラムを作成する

Step1 ルーレットみたいにするには？

```
10 INPUT N
```

```
50 OUT 1<<N
```

```
70 WAIT 60
```

```
80 LET N,N+1
```

```
100 GOTO 50
```

SAVE0



RUN



N	LED						N *
	6	5	4	3	2	1	
0	○	○	○	○	○	●	0+1 = 1
1	○	○	○	○	●	○	1+1 = 2
2	○	○	○	●	○	○	2+1 = 3
3	○	○	●	○	○	○	3+1 = 4
4	○	●	○	○	○	○	4+1 = 5
5	●	○	○	○	○	○	5+1 = 6
0	○	○	○	○	○	●	0+1 = 1

Nの値を0から5までを繰り返えせばいいので・・・。

Nを1ずつ足す命令は、80 LET N,N+1

Step2 くるくる回すには？

もし、Nが？になったら、Nは0にする IF ? THEN LET N,0
これで、RUNでルーレットスタート、ESCキーでストップする、ルーレットの完成

```
10 INPUT N
```

```
50 OUT 1<<N
```

```
70 WAIT 60
```

```
80 LET N,N+1
```

```
90 IF N=  THEN LET N,0
```

```
100 GOTO 50
```

SAVE0



LIST



RUN



※PRINT(?)命令でNの値と、LEDの光る場所を確認して みよう

```
75 PRINT N,BIN$(1<<N,6)
```

Step3 ボタンで スタート、ストップ！ するには？

ウォーミングアップのボタンを参考に、
スタート、ストップボタンを決めましょう。

```
10 LET N,0  
30 IF IN(□)=1 GOTO 30  
50 OUT 1<<N  
70 WAIT 6  
80 LET N,N+1  
90 IF N=6 THEN LET N,0  
100 IF IN(□)=1 GOTO 50  
300 BEEP 5,30  
310 GOTO 30
```

STOP START



IN2

IN1

SAVE0



LIST



RUN



10行目

INPUT命令の代わりに、Nの値をあらかじめ決めておきます。
このような処理のことを初期化(しょきか)といいます。

30行目

スタートボタンが押されるのを待つ命令を加えます。

70行目

回転速度を速くします。

100行目

ストップボタンを押されるまでプログラムを繰り返す命令を加えます。

300行目以降

ストップボタンが押された後の処理です。
ここでは、ルーレットが止まったことがわかるように、
BEEP命令を追加しました。

310行目

GOTO命令で、ボタン待ちを処理に戻ります。

これで完成！！！！

Step 4 プログラムをどんどん改造しよう！

※BEEP音を追加。いろいろ音色を変えて改造しよう！

60 BEEP

60 BEEP 100

60 BEEP 2

60 BEEP15:WAIT1:BEEP5



※速度を調整するには、WAITコマンドの値を変更する。

70 WAIT 1

※乱数でNの値を初期化

10 LET N,RND(6)

※ルーレットを逆回転させる。

50 OUT 1<<(5 -N)

または

50 OUT 32>>N

自分のオリジナル改造を
してみましょう。

解らない時は講師の先生
に聞いてみましょう。

♥ 6灯LEDルーレット

```
1 'HEART*6
10 LET N,2
20 OUT 1<<N
30 IF IN(1)=1 GOTO 30
40 LET W,1
50 OUT 1<<N
60 BEEP 100
70 WAIT W
75 PRINT N,BIN$(1<<N,6),IN(4),IN(3),IN(2),IN(1)
80 LET N,N+1
90 IF N=6 THEN LET N,0
100 IF IN(4)=1 GOTO 50
200 OUT 11,0
210 LET W,W+2
220 IF W<30 GOTO 50
230 OUT 11,-1
300 BEEP 5,30
310 GOTO 30
```

【遊び方】

1. 押しボタンIN1を押すとルーレットが回転します。
2. 押しボタンIN4を押すと、徐々に回転速度が落ちてやがて止まります。
3. 止まった箇所のLEDを点灯し、音を鳴らして当たりを知らせます。
4. 1.から繰り返します。

SAVE0



LIST



RUN



【プログラム説明】

10行目 変数Nを初期化。最初にLEDを点灯する位置をいれる(0~5)。
20行目 N番目のLEDを点灯する。
30行目 ボタン1が押されるのを待つ。ここがスタートボタン待ち。
40行目 変数Wを初期化。WにはLEDの点灯時間が入る。1が最も早い。
50行目 N番目のLEDを点灯する。
60行目 音を鳴らす。LEDが1つ動くごとに鳴る。
70行目 LED点灯待ち時間。W値の分だけ点灯。
80行目 Nの値を1つプラス。
90行目 もし、Nが6なら Nに0を入れる。
これでルーレットが一周したことになる。
100行目 ボタン4が押されていないならば、GOTO50へ飛んで次のLEDを点灯。
200行目 IN4をOUTポート(OUT11)に変更して0を出力。
こうするとIN4の値は常に0となり、
ボタンを押したままと同じ状態になる。
210行目 WAIT値のWを少し増やす。だんだんゆっくりになる。
220行目 WAIT値Wが30以下なら、GOTO50
230行目 IN4をOUTポートからINポートに戻す。
300行目 ルーレットが停止した合図(BEEP)を鳴らす。

【応用】

・逆回りに光らせたいときは、
20行目と50行目をつぎのようにします。
OUT 1<<(5-N) または OUT 32>>N

・220行目をつぎのようにかえると、
ルーレットの止まり方が毎回変わります。
IF W<20+R GOTO 50 ELSE LET R,RND(20)

HEART★6 タイマー



6分タイマーです。

タイマーは、1分単位で時間をセットし、時間が来たらアラームが鳴ります。
SAVE0にプログラムを保存しておけば、BTNを押しながら電源を入れると、タイマーのプログラムが自動スタートします。

```
1'HEART*6タイマー
10 LET[1],4,6,7,39,55,63:C=3:CLO
20 @SET
30 OUT [C]
40 GSB @SW(S): IF S CONT
50 GSB @SW(S): IF !S CONT
60 BEEP 20
70 C=C-(S=8&&C>1)+(S=1&&C<6)
80 IF S<>4 GOTO @SET
90 CLT:BEEP5,100
100 @TIMER
110 OUT 1<<(TICK())/60)
120 IF TICK()/15%2 THEN OUT [C-TICK()/3600]
130 GSB @SW(S): IF S=2 BEEP 50,50:GOTO @SET
140 IF TICK()/C*3600 GOTO @TIMER
150 @ALM
160 PLAY"O5$B8C6"
170 OUT TICK()/10%2*63:GSB @SW(S): IF !S CONT
180 BEEP
190 GOTO @SET

1000 @SW(S):S=!IN(4)*8+!IN(3)*4+!IN(2)*2+!IN(1):RTN
```

チャレンジ

1. プログラムスタート時はタイマーセットモードです。

セット中の時間分だけLEDが点灯します。
ボタンを操作して時間をセットします。

IN1(SW1) -1分減らす(最小1分)

IN4(SW4) +1分増やす(最大6分)

IN3(SW3) セット完了&スタート


2. タイマー動作中は、1秒毎にLEDの光が一周し、残り時間の数だけLEDが点滅します。
IN2(SW2)ボタンを押すとキャンセルとなり、1.のタイマーセットモードになります。

3. 時間が来たら、全LEDが点滅し、アラームが鳴ります。

どれかボタンを押すと、1.のタイマーセットモードになります。

コマンドをおぼえよう !!

プログラムをリストするには

LIST 

範囲を指定してリスト
LIST 10,100

プログラムをすべて削除するには

NEW 

プログラムを実行するには

RUN 

画面表示をすべて消します（クリア スクリーン）

CLS 

プログラムをメモリーに記憶させるには

SAVE 

(0~3)の4個のメモリーがあります

プログラムの行削除は行番号だけを打ちます

10 

プログラムをメモリーから読みだすには

LOAD 

メモリーの内容は電源を切っても記憶しています

実行しているプログラムを停止するには



キーを押します

エスケープと読みます
(Escape: 逃れる、抜け出す、脱出する)の省略型です

メモリーの内容をみるには

FILES 

コマンドはキー入力だけでなくファンクションキーでも代用できるよ



F1	F2	F3	F4	F5	F6	F7	F8	F9
CLS	LOAD	SAVE	LIST	RUN	?FREE()	OUT0	VIDEO1	FILES

Dec	Hex	Bin
0	#00	0000
1	#01	0001
2	#02	0010
3	#03	0011
4	#04	0100
5	#05	0101
6	#06	0110
7	#07	0111
8	#08	1000
9	#09	1001
10	#0A	1010
11	#0B	1011
12	#0C	1100
13	#0D	1101
14	#0E	1110
15	#0F	1111

4bit: 0~15
8bit: 0~255
(-128~127)
16bit: 0~65535
(-32768~32767)
*数える数値は16ビット範囲
小数値は使えません。

VIDEO1	KBD1
VIDEO2	EX1
IN1	KBD2
IN2	SOUND
IN3	ISP
IN4	RESET
VCC	GND
GND	VCC
OUT1	-
OUT2	-
OUT3	OUT5
OUT4	OUT6
BTN	TXD
LED	RXD

F1	F2	F3	F4	F5	F6	F7	F8	F9
CLS	LOAD	SAVE	LIST	RUN	?FREE()	OUT0	VIDEO1	FILES

IchigoJam BASIC 1.1 cheatsheet

*赤文字は省略可能を示す *青文字は説明文

演算の優先順位
高い () - ~ ! NOT * / % MOD << >> & ^ + - | = != < > <= >= AND OR 低い

●式/演算

【加算】数+数
【減算】数-数
【乗算】数*数
【除算】数/数
【剰余】数%数
【否定】NOT 式
【論理積】数&数
【論理和】数|数
【排他的論理和】数^数
【右シフト】数>>数
【左シフト】数<<数
【ビット反転】~数
【優先順位変更】(~)
式AND式 省略形:&&
式OR式 省略形:||

●代入/変数/配列変数

【数】 0~101 まで
LET 変数, 数 省略形: 変数 = 数
LET [数], 数 ... 数 n

●リセット/初期化

CLK キーバッファ消去
CLP パターン初期化
CLS 画面消去
CLT 時間をリセット
CLV 変数を消去し全て0に
SRND 数 乱数の種を設定

●キー入力/ボタン

BTN(UP/DOWN/RIGHT/LEFT/SPACE)
INKEY() リアルタイムキー入力
INPUT 文字列, 変数

●画面関係

LOCATE X 座標, Y 座標 省略形: LC
PRINT 数や文字列 省略形: ?
SCR(X 座標, Y 座標)
SCROLL 数 0: 上, 1: 右, 2: 下, 3: 左
VIDEO 数 1, 数 2

●関数

ABS(数)
ASC("文字")
BIN\$(数, 桁数)
CHR\$(数, ... 数 n)
DEC\$(数, 桁数)
HEX\$(数, 桁数)
RND(数)

●数値表記

123 10進数
(-32768~32767)
#E9 16進数(0~#FFF)
`1001 2進数

●定数

LEFT 左: 28
RIGHT 右: 29
UP 上: 30
DOWN 下: 31
SPACE 空白: 32

●条件判断/条件式

IF 数 THEN 次 ELSE 次 2
【等しい】数1=数2
【等しくない】数1<>数2
【小さい】数1<数2
【小さいか等しい】数1<=数2
【大きい】数1>数2
【大きいか等しい】数1>=数2

●移動/繰り返し/サブルーチン

FOR 変数 = 数 1 TO 数 2 STEP 数 3 ~ NEXT
GOSUB 行番号 省略形: GSB
GOTO 行番号
RETURN 省略形: RTN

●ハードウェア

ANA(数) 0~1023
BPS 通信速度 省略時: 115,200bps
IZCR(数1, 数2, 数3, 数4, 数5)
IZCW(数1, 数2, 数3, 数4, 数5)
IN(数) IN1-9 から入力
LED 数 0: 消灯, 1: 点灯
OUT 数 1, 数 2 OUT1-7 に出力
PWM 数 1, 数 2, 数 3
RESET リセット
SLEEP スリープ (ボタンを押すと復帰)
UART シリアル出力設定, シリアル受信設定
WAIT 数 60 で約 1 秒

●ファイル

FILE()
FILES 数 1, 数 2
LOAD 数
LRUN 数, 行番号
RUN
SAVE 数

●プログラム

CONT 再度実行する
END プログラムを終了
FREE() プログラムの残りメモリ数
LINE() 現在実行中の行番号
LIST 行番号 1, 行番号 2
NEW プログラムを消す
RENUM 数 1, 数 2
STOP 処理を中断する

●メモリ操作/マシン語

PEEK(アドレス)
POKE アドレス, 数 ... 数 n
USR(アドレス, 数)

●その他

HELP メモリマップを表示
REM 注釈 省略形: '
TICK() tick 時間 (1/60) を返す
VER() バージョン番号を返す

#0000	メモリマップ
#0700	文字パターン (#00~#DF)
#0800	PCGパターン (#E0~#FF)
#0900	配列変数・変数
#0C00	画面 (32 文字 × 24 行)
#1001	プログラムリスト
#1002	キーが押されたビット
#1003	キーバッファ格納数 (最大 14)
#1004~F	キーバッファ

●音楽/サウンド

BEEP 周期, 長さ BEEP を鳴らす
PLAY [MML] MML なしで演奏停止
SOUND() 再生中なら 1 を返す
TEMPO テンポ テンポを指定

■MML (Music Macro Language)

【音】音 (C D E F G A B R)
【音]n 音長 (を付けると 1.5 倍長)
【音]+ 半音上げる
【音]- 半音下げる
Tn テンポ (初期値: 120)
Ln デフォルトの音長 (初期値: 4)
On オクターブ指定 (1~5)
> 1 オクターブ上げる
< 1 オクターブ下げる
\$ 以後の MML を繰り返す
Nn 音の高さを指定
(音長で指定可能な値: 1, 2, 3, 4, 8, 16, 32)

●制御 (コントロール) コード

08(#08) バックスペース (後退)
13(#0D) リターン
14(#0E) インサート (挿入)
127(#7F) デリート (削除)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	:
2	;	<	=	>	?@	[\]	^	_	`	{		}	~	DEL
3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
5	q	r	s	t	u	v	w	x	y	z	[\]	^	_	`
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	[\]	^	_
8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
9	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
A	q	r	s	t	u	v	w	x	y	z	[\]	^	_	`
B	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
C	p	q	r	s	t	u	v	w	x	y	z	[\]	^	_
D	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
F	q	r	s	t	u	v	w	x	y	z	[\]	^	_	`

1 文字は 8 × 8 ドットで構成