

# Arduino はやみ表

ver1.2 by 武蔵野電波



## Blink サンプル

```
const int ledPin = 13;
void setup() { // 1度だけ実行される
  pinMode(ledPin, OUTPUT);
}
void loop() { // 繰り返し実行される
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

## 制御構造

```
if (x > 8) {...} else {...}
for (int i = 0; i < 8; i++) {...}
while (x < 8) {...}
do {...} while (x < 8);
continue; ループの残りの部分を飛び越す
break; 処理を中止して抜ける
return x; 関数から抜けて値 x を返す
switch (x) {
  case 1:
    break;
  case 2:
    break;
  default:
}
// コメント
/* 長さが自由なコメント */
#define LEDPIN 12
#include <EEPROM.h>
```



Mega

## コメントと特別な命令

```
// 1行ずつのコメント
/* 長さが自由なコメント */
#define LEDPIN 12
#include <EEPROM.h>
```

## 演算子

```
x = y + 3; y = x - 3;
x = y * 5; y = x / 5;
a = b % 8; 8で割った余りを求める
```

```
x == y 等しい x != y 等しくない
x < y x > y
x <= y x >= y
```

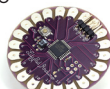
```
i++ 評価して加算 ++i 加算して評価
i-- 評価して減算 --i 減算して評価
x += 2; は x = x + 2; と同じ
```

```
&& どちらも真なら真 ((x < y) && (y < z))
|| どちらかが真なら真 ((x == 1) || (y != 1))
! 否定 if (!x) {...}
```

```
x &= B11111100; マスク (AND)
x |= B00000011; セット (OR)
z = x ^ y; 排他的論理和 (XOR)
y = ~x; 否定 (NOT)
y = x << 2; 左シフト y = x >> 2; 右シフト
```

## 型

void	
boolean	真 true か偽 false
char	-128 ~ 127
unsigned char	0 ~ 255
byte	0 ~ 255
int	-32768 ~ 32767
unsigned int	0 ~ 65535
word	0 ~ 65535
long	-2147483648 ~ 2147483647
unsigned long	0 ~ 4294967295
float	-3.4028235E+38 ~ 3.4028235E+38
double	-3.4028235E+38 ~ 3.4028235E+38



LilyPad

## 文字列

```
char str[] = "hello"; 配列として初期化
str[0] = 'H'; 1文字目をHに変更
"Hello\tworld!\r\n" タブと改行 (CR+LF)
print( F("Hello") ) Flashメモリを使用
```

## 定数と数値表現

HIGH | LOW デジタル入出力の値  
INPUT | OUTPUT デジタル入出力の向き  
true | false 論理値 (真と偽)  
170 十進数 0252 八進数  
0xAA 十六進数 B10101010 二進数  
10U 符号なし  
20L long 30UL 符号なし long  
10.0 浮動小数点数  
2.4e5 240000.0

## 配列

```
int array[5]; 要素を5個持つ配列
array[0] = 2; ひとつめの要素に代入
int pins[] = {2, 4, 8, 6};
sizeof(pins)/sizeof(pins[0]) 要素の数
```

## 型宣言で使うキーワード

```
const float pi = 3.14;
volatile char buf;
static int result;
```

## デジタル入出力

```
pinMode(pin, [INPUT|OUTPUT])
digitalWrite(pin, [HIGH|LOW])
int x = digitalRead(pin);
内蔵プルアップ抵抗を有効にする
pinMode(pin, INPUT);
digitalWrite(pin, HIGH);
```

## アナログ入出力

```
int x = analogRead(pin);
analogReference([DEFAULT|INTERNAL|
EXTERNAL]) デフォルトは電源電圧
analogWrite(pin, x) xは0 ~ 255
```

## その他の入出力

```
shiftOut(dataPin, clockPin,
[MSBFIRST|LSBFIRST], value)
shiftIn(dataPin, clockPin,
[MSBFIRST|LSBFIRST])
pulseIn(pin, [HIGH|LOW])
tone(pin, freq) 周波数はヘルツ (Hz) で指定
tone(3, 440, 90); 90ミリ秒間だけ鳴らす
noTone(pin)
```

## 時間

```
millis() 起動からの経過時間 (ミリ秒)
micros() 起動からの経過時間 (マイクロ秒)
delay(250); 250ミリ秒間停止
delayMicroseconds(250); 250マイクロ秒
```

## 乱数

```
randomSeed(analogRead(0)); 初期化の例
long x = random(max); max-1までの整数
long x = random(min,max); 最小値を指定
```

## 数学的な関数

```
min(x, y) max(x, y)
abs(x)
sqrt(x) pow(base, exponent)
sin(rad) cos(rad) tan(rad)
constrain(x, min, max)
map(x, fromL, fromH, toL, toH)
```

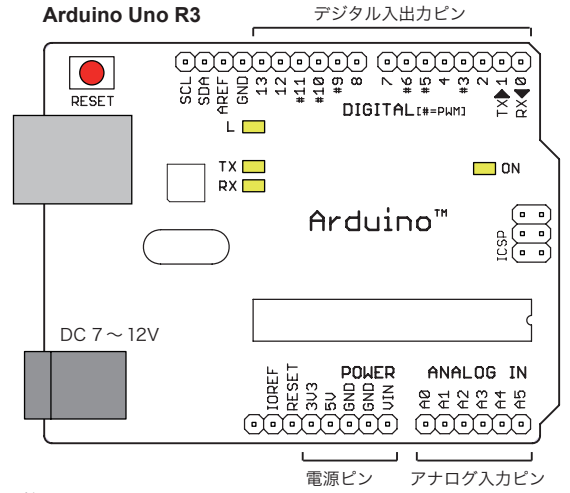
## ビットとバイトの処理

```
lowByte(x) highByte(x)
bitRead(x, n) bitWrite(x, n, bit)
bitSet(x, n) bitClear(x, n)
bit(n) (1 << (n)) と同じ処理
```

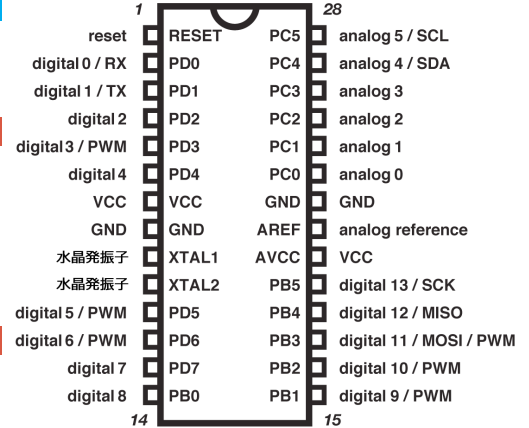
## 外部割り込み

```
attachInterrupt([0|1], function,
[LOW|CHANGE|RISING|FALLING])
detachInterrupt([0|1])
noInterrupts() 割り込みの一時停止
interrupts() 止めた割り込みの再スタート
```

## Arduino Uno R3



## ATmega168/ATmega328P



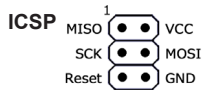
## FTDI USB IF

black	GND
brown	CTS#
red	VCC
orange	TXD
yellow	RXD
green	RTS#

Sparkfun FTDI Basic の green は DTR 端子

## メモリ容量の比較

	mega168	mega328	mega1280
Flash	16KB	32KB	128KB
SRAM	1KB	2KB	8KB
EEPROM	512B	1KB	4KB



## シリアル通信

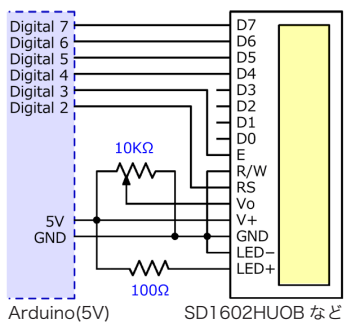
```
Serial.begin(9600); 初期化 9600bps
受信した 1 バイトを 10 進数で送り返す例
if (Serial.available() > 0) {
  buf = Serial.read(); // 1文字読む
  Serial.print("I received: ");
  Serial.println(buf, DEC);
}
```

## String クラス

```
String s1 = "Hello";
print( s1 + s2 ); 文字列の連結
if ( s1 == s2 ) 文字列の比較
```

## キャラクタ液晶ディスプレイ

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
lcd.begin(16,2); LCDの桁数と行数を指定
lcd.setCursor(10,1); カーソル位置を指定
lcd.print("Hello world!");
lcd.clear(); 画面をクリアしカーソルは左上
```



Arduino(5V)

SD1602HUOB など



Nano