

2014年6月28日:草稿
2014年7月4日:初稿
2014年9月10日:2稿

ロボットをつくろうWS 2014年9月24日（水）

徳島大学大学院ソシオテクノサイエンス研究部
技術専門職員 辻 明典

連絡先：

770-8506 徳島市南常三島町2-1

TEL/FAX：088-656-7485

E-mail: : a-tsuji@is.tokushima-u.ac.jp

本日の予定

- 1 WSの概要
 - 2 WSで製作するロボット
 - 3 配付部品の確認
 - ー半田付けー
 - 4 マイコンについて
 - 5 Arduino開発環境のインストール
 - 6 マイコンボードの動作確認
 - 7 マイコンボードについて
- 付録：テープLEDの光らせ方

1 WSの概要

WSの概要

ロボットをつくらうワークショップ

時間：10時～12時，13時～

スケジュール：

- ① 9/24（水）10時～12時，13時30分～16時00分
WS概要，WSロボットの概要，半田付け，マイコンボード製作
Arduinoの開発環境，マイコンボードのテスト
- ② 9/25（木）10時～12時，13時30分～16時30分
モータボード製作
プログラム1（LED，ブザー，スイッチ，Bluetooth無線）
省電力プログラム（PIR）
- ③ 9/26（金）10時～12時
プログラム2（フォトインタラプタ，サーボモータ）
ロボットの制御（ライントレース，PID制御）

WSの概要

準備する物：

- ① 半田ごて道具一式
- ② ノートパソコン
- ③ リモコン（赤外線式）

WS (1日目)

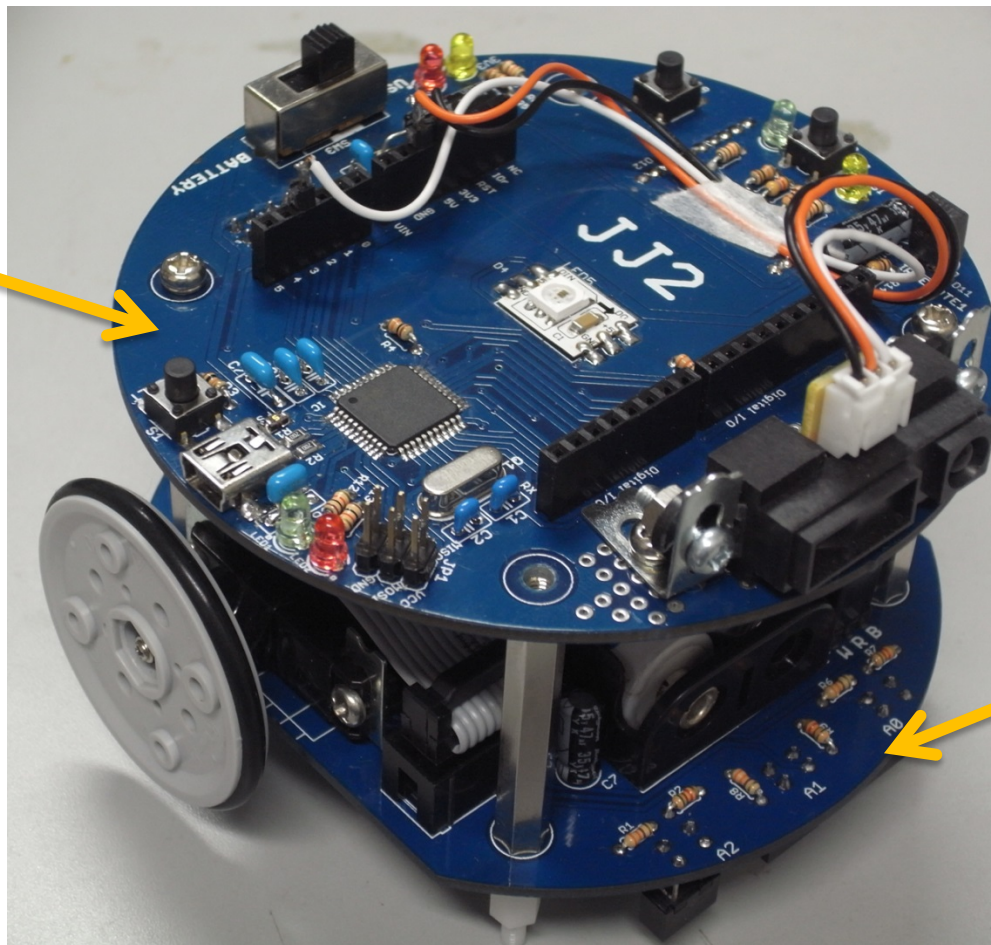
- 1 WSの概要
- 2 WSで製作するロボット
- 3 配付部品の確認
 - ー半田付けー
- 4 マイコンについて
- 5 Arduino開発環境のインストール
- 6 マイコンボードの動作確認
- 7 マイコンボードについて

2 WSで製作するロボット

WSで製作するロボットの外観

二輪移動型ロボット (JJ)

Arduino互換
マイコンボード



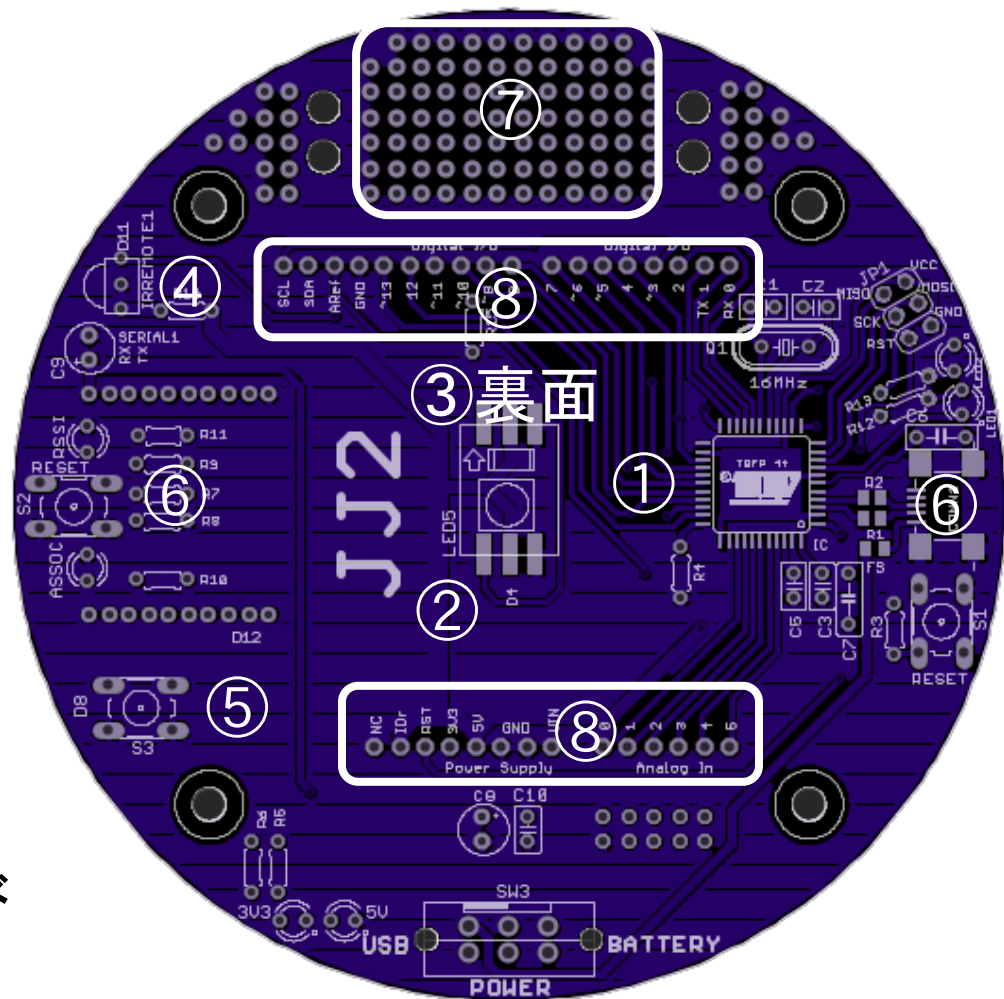
モータ制御ボード

Arduino互換マイコンボード

JJに出来ること

- ① マイコンのプログラム
- ② フルカラーLEDの点灯
- ③ スピーカを鳴らす
- ④ リモコンによる操作
- ⑤ プログラム起動スイッチ
- ⑥ 有線・無線通信
- ⑦ センサ増設など拡張エリア
- ⑧ Arduino互換ピンヘッダ

Arduino Leonardo互換ボード
として使用可能



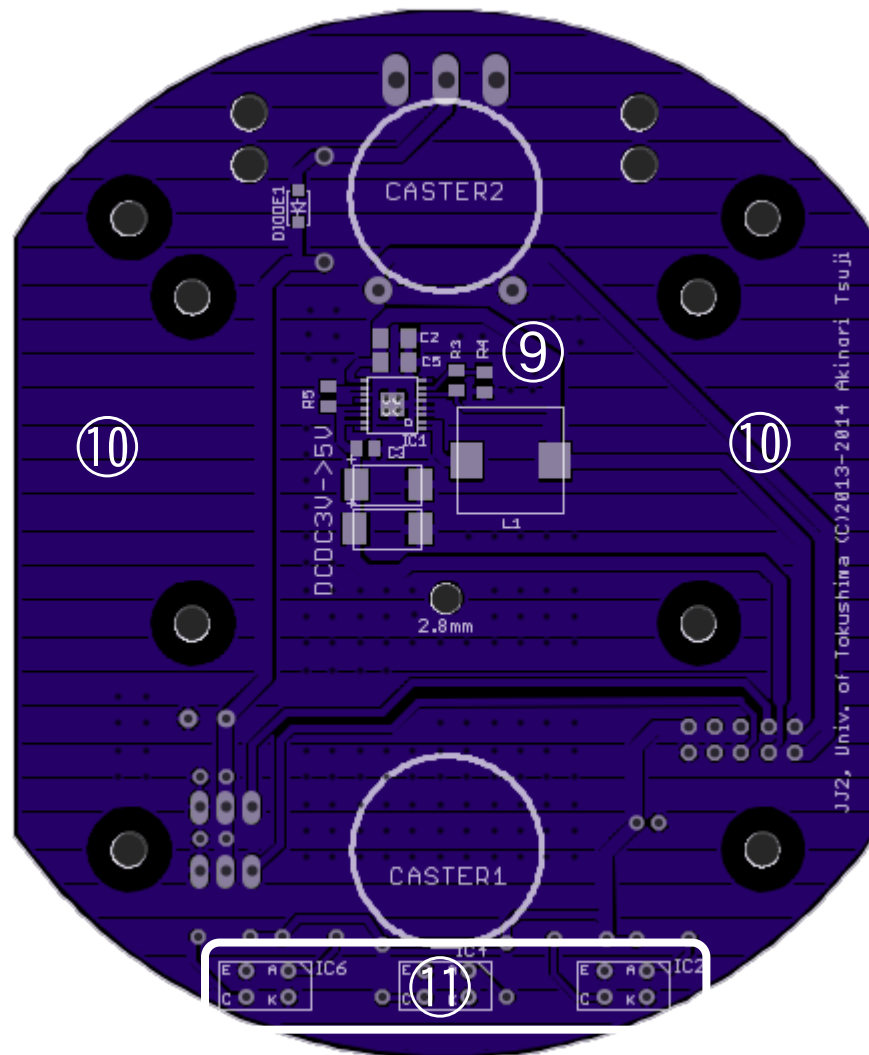
モータ制御ボード

JJに出来ること（つづき）

- ⑨ 電源供給（単3電池2本）
- ⑩ サーボモータ（2個）の制御
- ⑪ フォトインタラプタ（3個）によるライントレース

ロボット製作の目的

- ・ロボットのしくみを知る。
- ・プログラミングを習得する。
- ・センサの制御を知る。
- ・マイコンの使い道を知る。
- ・アイデア次第でロボット以外にも。



3 配付部品の確認

配付部品の確認

2014(平成26年度)ロボットをつくろうWS部品表 (別紙)

- ・ 部品

 - Arduino互換マイコンボード基板 (1枚)

 - モータ制御基板 (1枚)

 - USBケーブル (1本)

 - 部品1式

- ・ DVDの内容

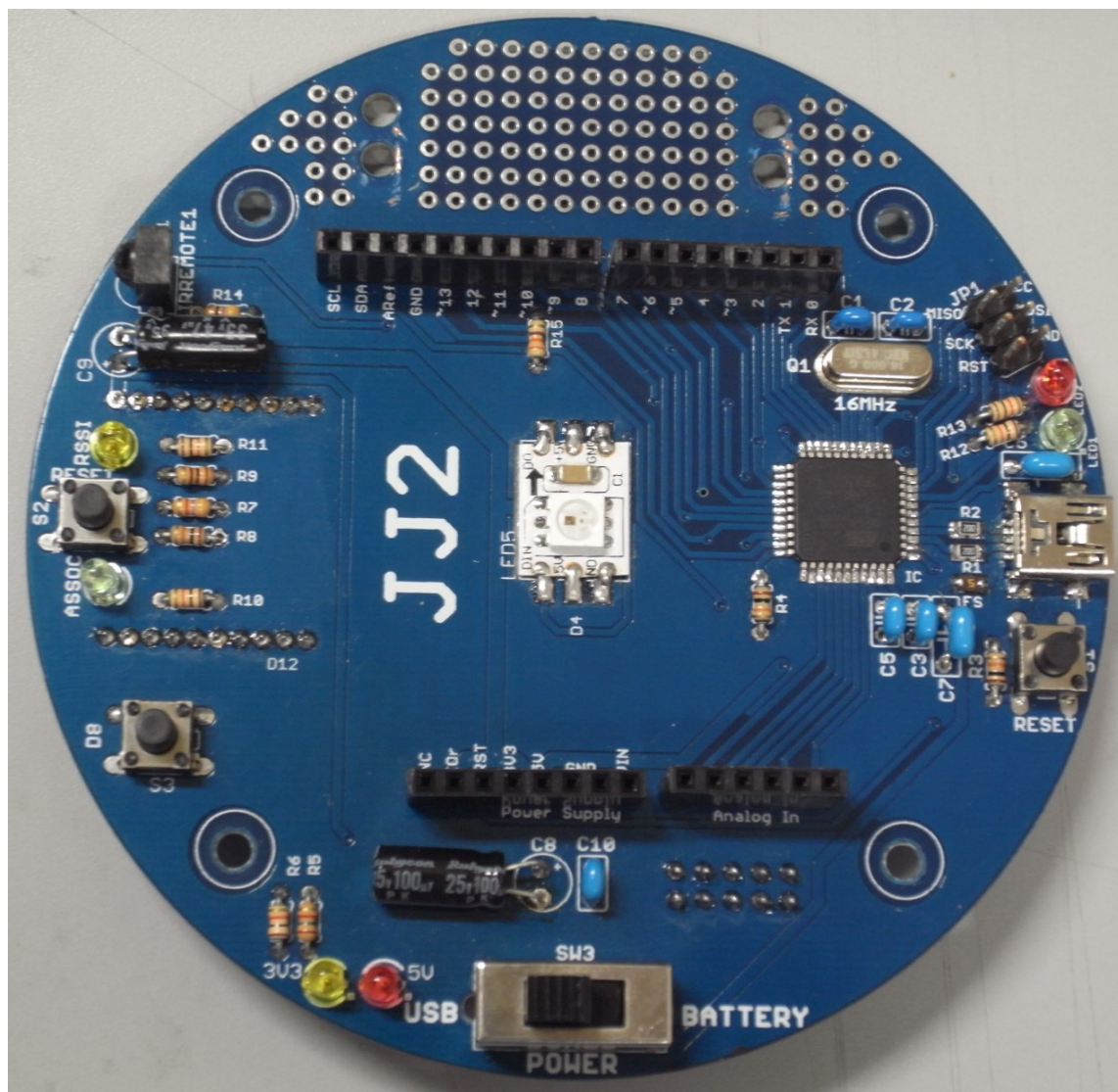
 - ソフトウェア・・・Arduino他

 - WS資料

 - データシート・・・各部品のデータシート (仕様)

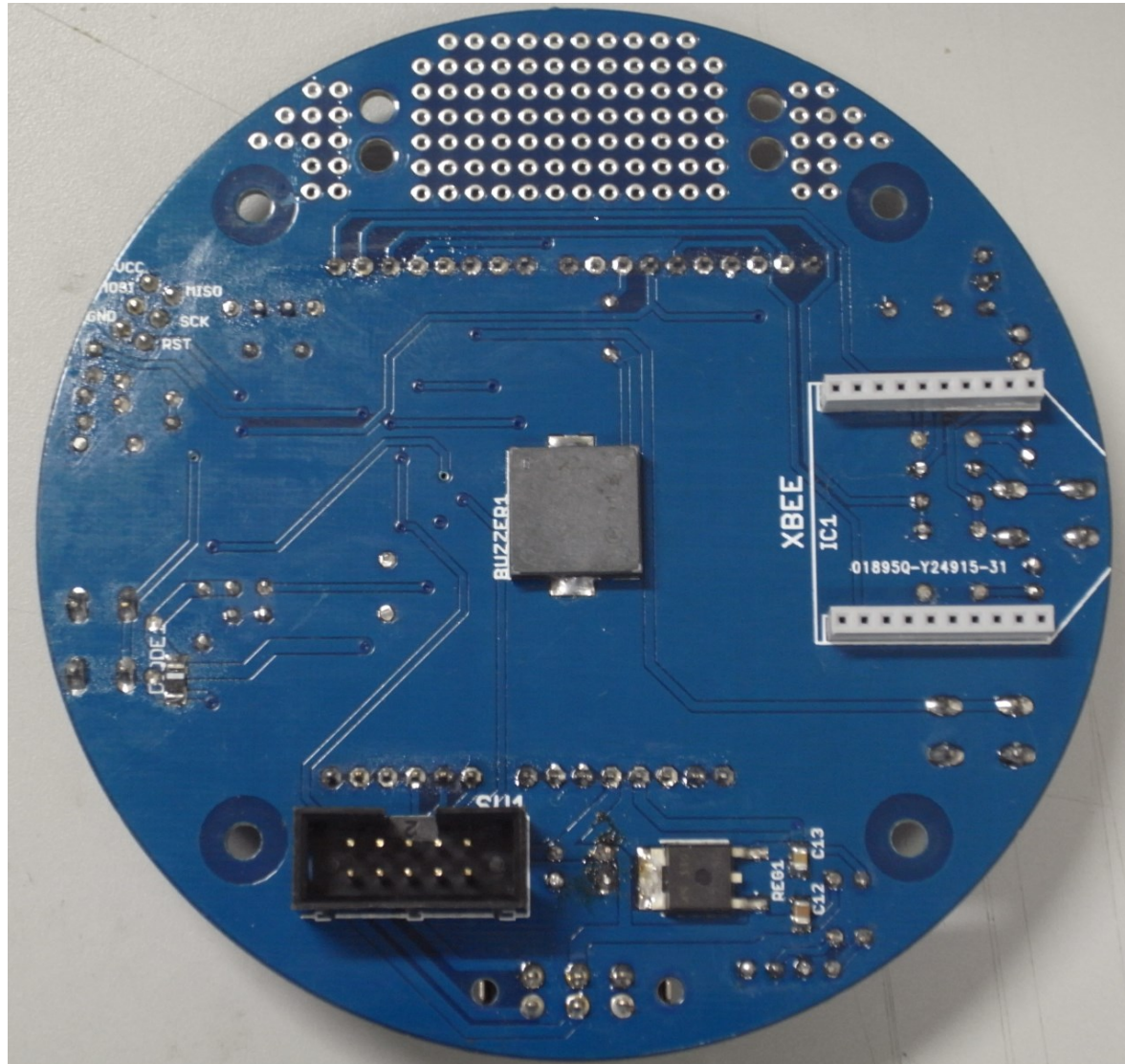
半田付け（別紙参照）

マイコンボードできあがり (表)



Jul 19, 2014, The University of Tokushima,
Akinori Tsuji

マイコンボードできあがり（裏）



4 マイコンについて

マイコン

マイコンとは？

マイクロコントローラ，マイクロコンピュータの略

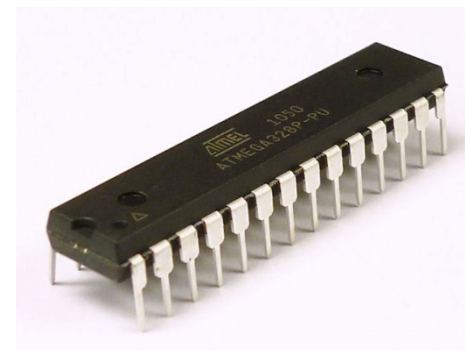
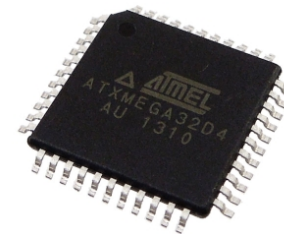
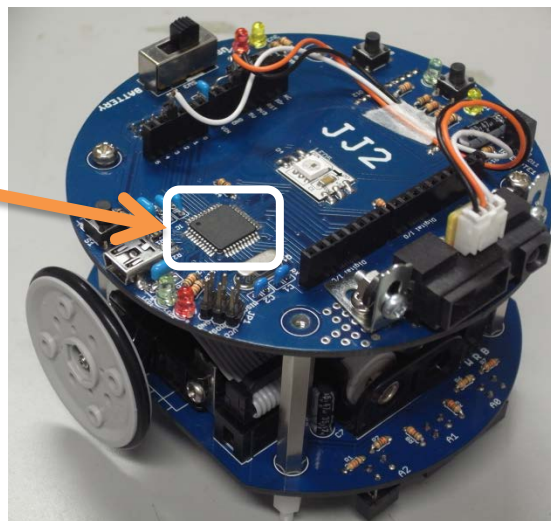
マイコンの必須機能：

プログラム，演算，入出力，メモリ

本WSでのマイコンの役割：

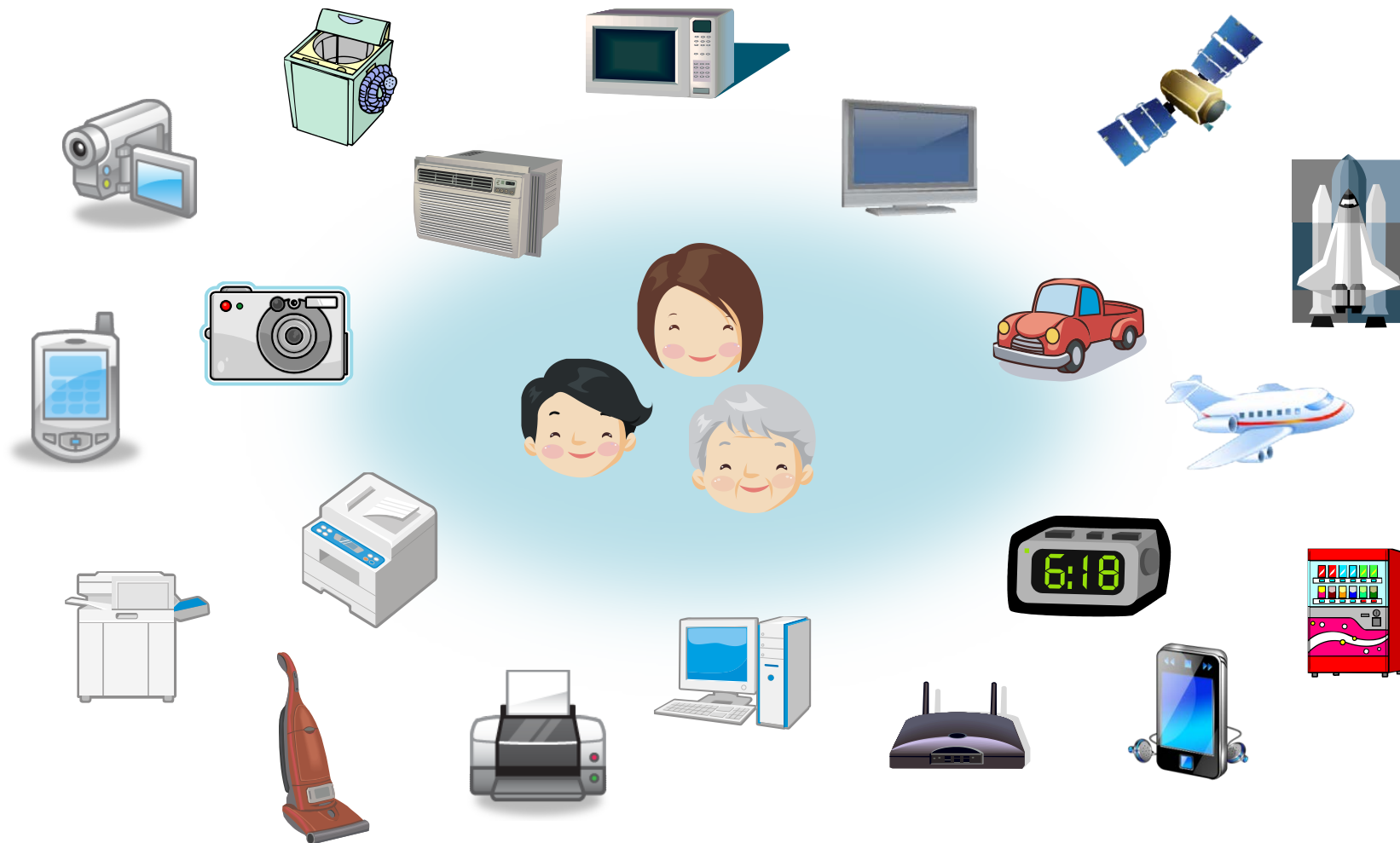
ロボットの頭脳

マイコン

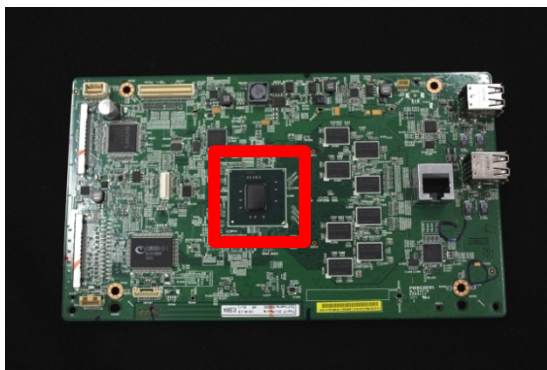


いろいろなマイコン

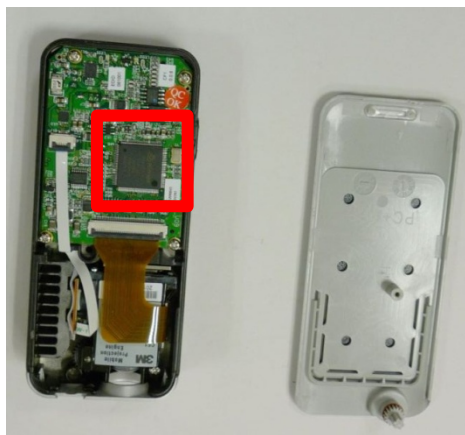
マイコンを探そう



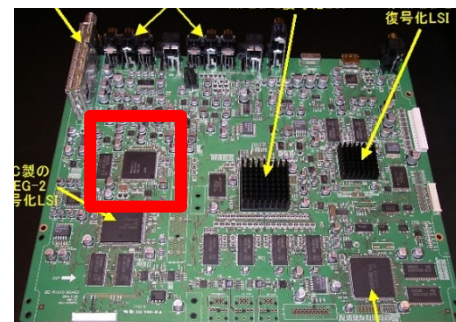
製品の中のマイコン



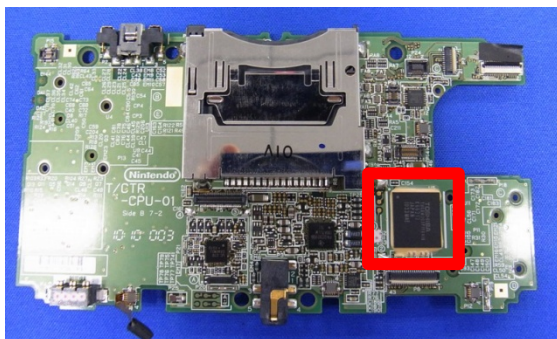
デジタルテレビ



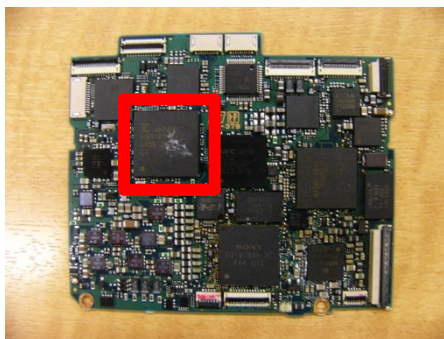
プロジェクタ



ブルーレイレコーダ



携帯ゲーム機



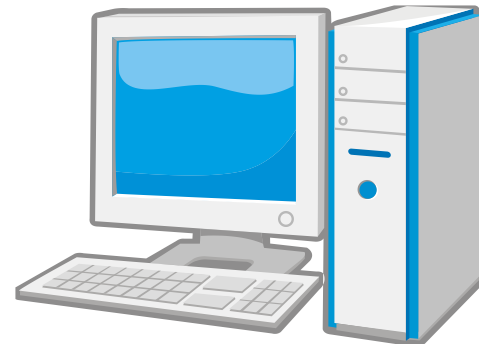
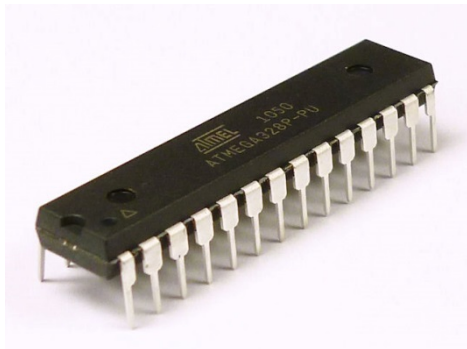
デジタルビデオカメラ

日系BPより写真転載

マイコンの特長

パソコンに使用される汎用プロセッサと比較

- ・用途：特定（単一の目的）
- ・処理速度：遅い
- ・メモリ：少い
- ・入出力：多い
- ・センサ：直接接続可能
- ・消費電力：低い，バッテリー動作可能
- ・量産コスト：安い



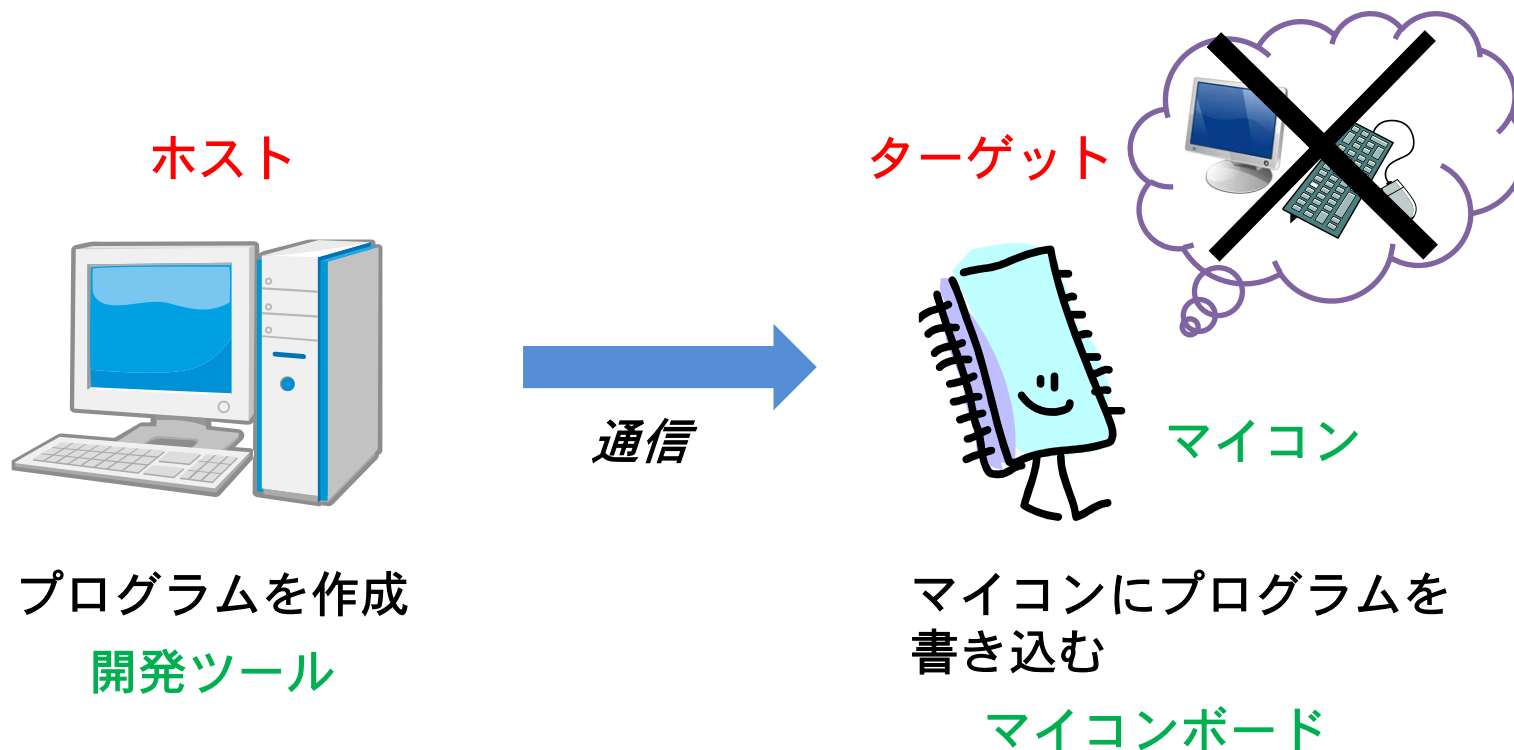
5 Arduino開発環境のインストール

一般的なマイコンの開発（クロス開発手法）

マイコンには、キーボードもディスプレイもない

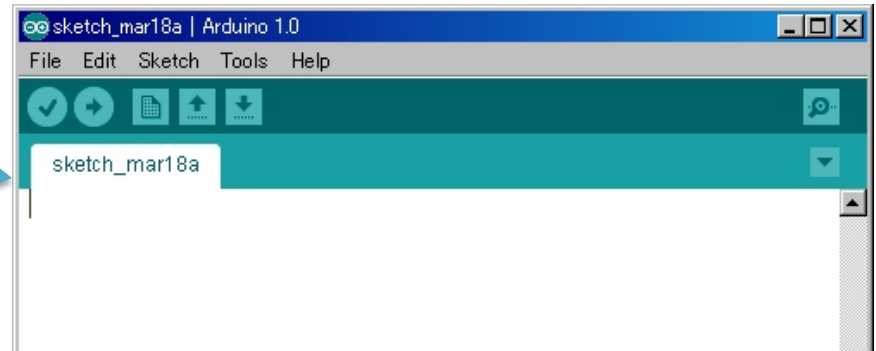
→ クロス開発環境

パソコン上でプログラムを作成，マイコンに書き込む



クロス開発に必要なもの

- パソコン
開発ツール
Arduino開発環境

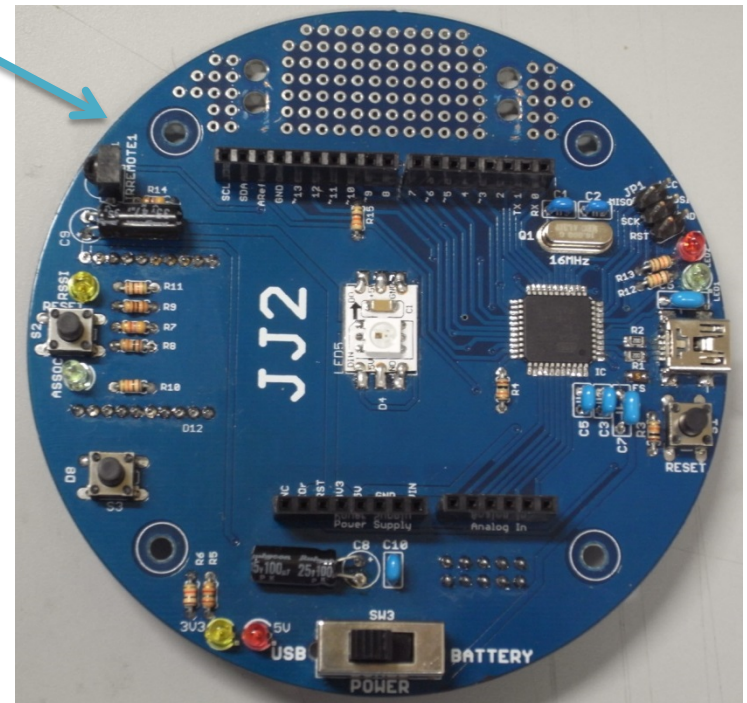
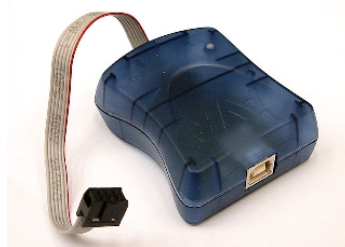


- マイコンボード
Arduino Leonardo互換
マイコン : ATmega 32u4

- USBケーブル
パソコンと接続



- ICEデバッガ
マイコンに直接プログラムを
書き込む・デバッグする



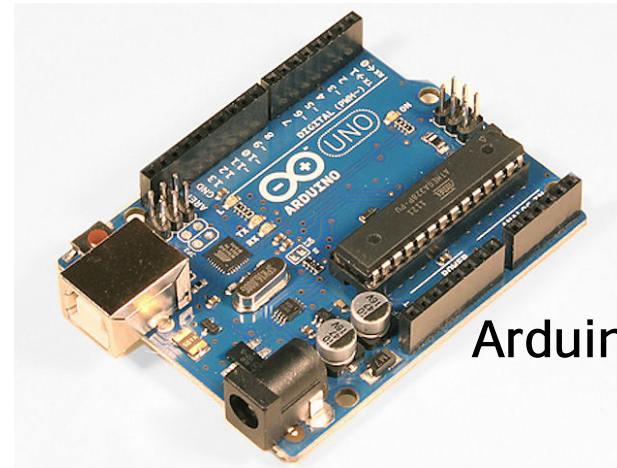
Arduinoとは

Arduino (アルデュイーノ)

電気・電子工学を専門としない人向けの教育用マイコンボード

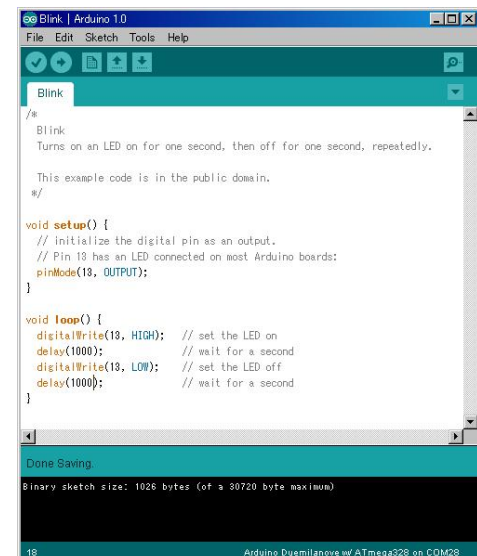
特長：

- ・ **オープンソースハードウェア**
 - ー 設計図が公開
- ・ **簡単にマイコンの開発が可能**
 - ー スケッチ=プログラム
- ・ **試作（プロトタイプ）が容易**
 - ー Arduino共通ピンソケット
 - ー 拡張ボード（シールド）
- ・ **統合開発環境**
 - ー エディタ
 - ー マイコン用のコンパイラ，アセンブラ，リンカを含む
 - ー シリアル通信モニタ



Arduino UNO

Arduino 開発環境



```
Arduino IDE - Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);           // wait for a second
}

Done Saving
Binary sketch size: 1026 bytes (of a 30720 byte maximum)
18 Arduino Duemilanove w/ ATmega328 on COM28
```


Arduinoのインストール条件

OS

Windows 8, 7, XP, Vista, MacOS, Linux

ディスク空き容量

1 GB以上

メモリ

1GB以上

USBポート

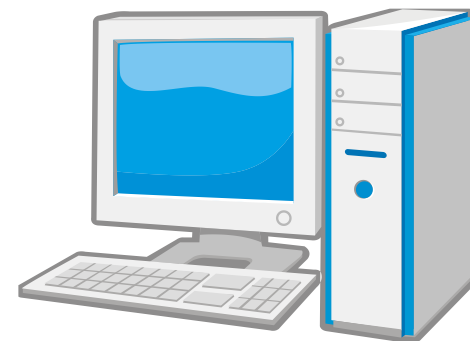
1個（マイコンボード接続用）

DVD/CD-ROMドライブまたはUSBメモリ

開発環境インストール

ソフトウェア（Arduino）

WEBサイトからダウンロード



<http://www.arduino.cc/>

Arduinoのインストール

1. DVD（または、USBメモリ）を入れる。
2. コンピュータからDVDドライブを開く。
3. **Software¥install** フォルダを開く。
4. **install.bat** をダブルクリックする。
5. インストールが完了するまで待つ。
インストール中、コマンドプロンプトを閉じない。
6. デスクトップにArduinoのアイコンが作成される。



Arduinoのアイコン

※ スケッチ（プログラム）は、
ライブラリ→ドキュメント→Arduino 以下にコピーする。

※ ライブラリ（FastLED, IRremote）
ライブラリ→ドキュメント→Arduino→libraries 以下に
展開したものをコピーする。

6 マイコンボードの動作確認

電源スイッチをOFF

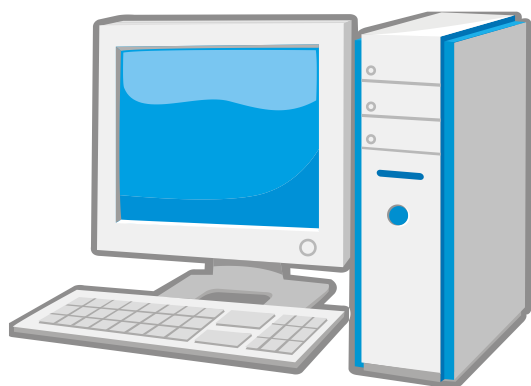
1. 電源スイッチをBATTERY側に切り替え（電源OFF）



電源スイッチ（BATTERY側にスライド）

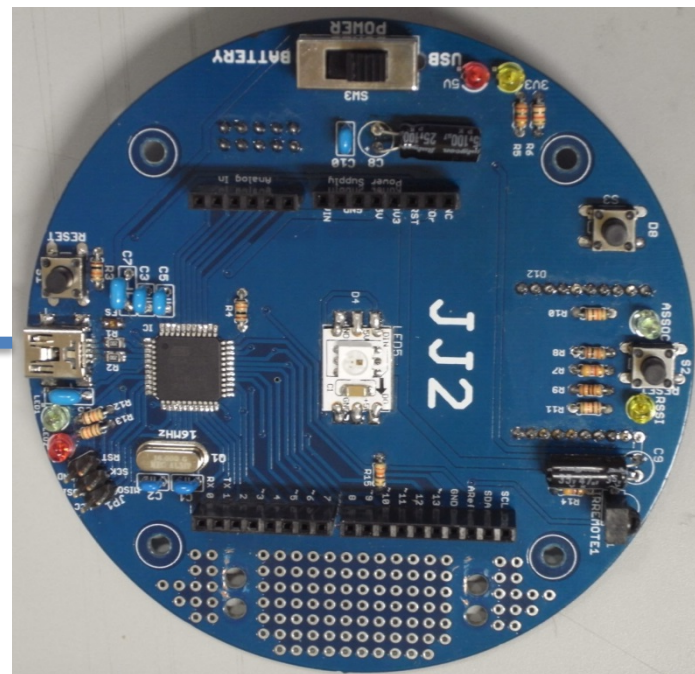
パソコンとJJマイコンボードの接続

2. パソコンとJJマイコンボードをUSBケーブルで接続



パソコン

USBケーブル

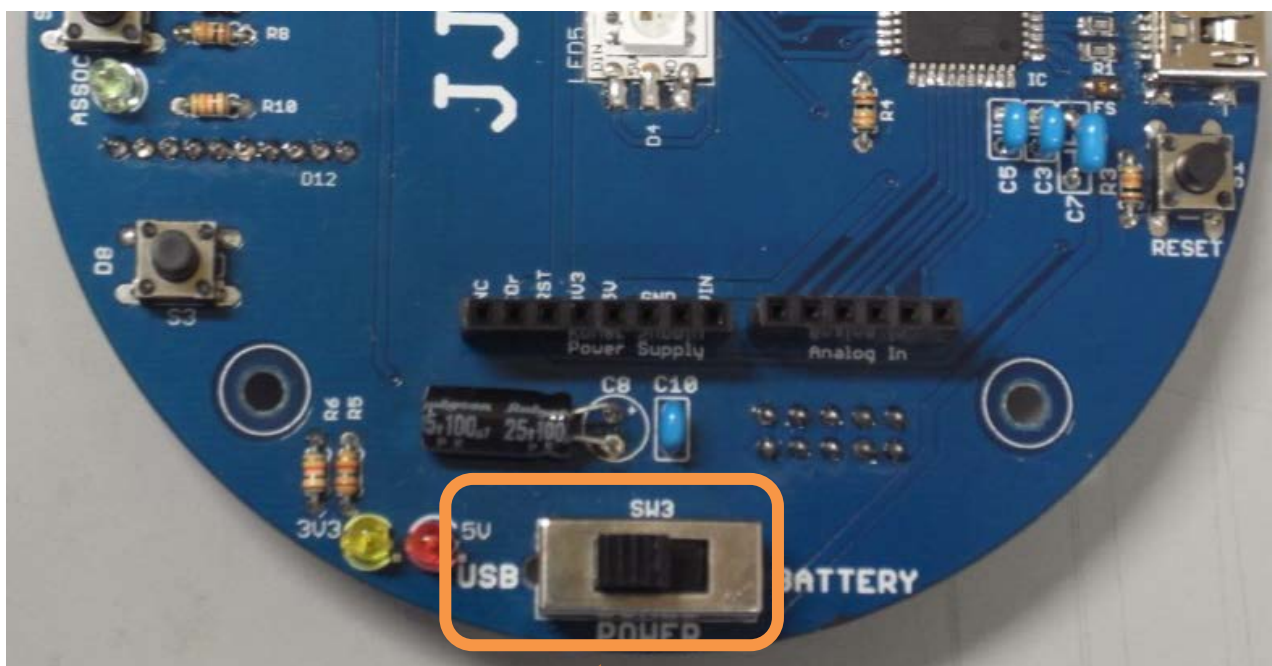


JJマイコンボード

電源スイッチをON

3. 電源スイッチをUSBに切り替え（電源ON）

※ 初回，USBドライバがインストールされるので完了まで待つ。

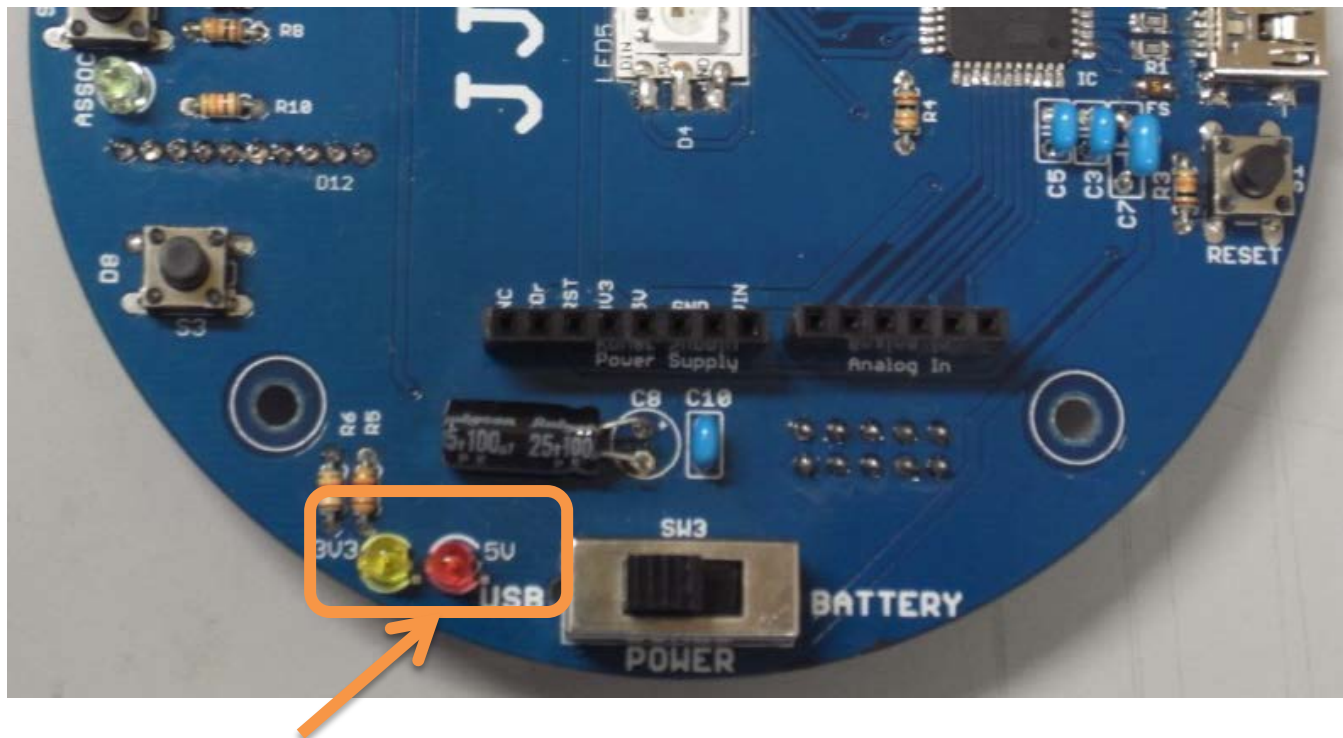


電源スイッチ（USB側にスライド）

電源LEDの確認

4. 電源を入れると，電源LEDが「必ず」点灯

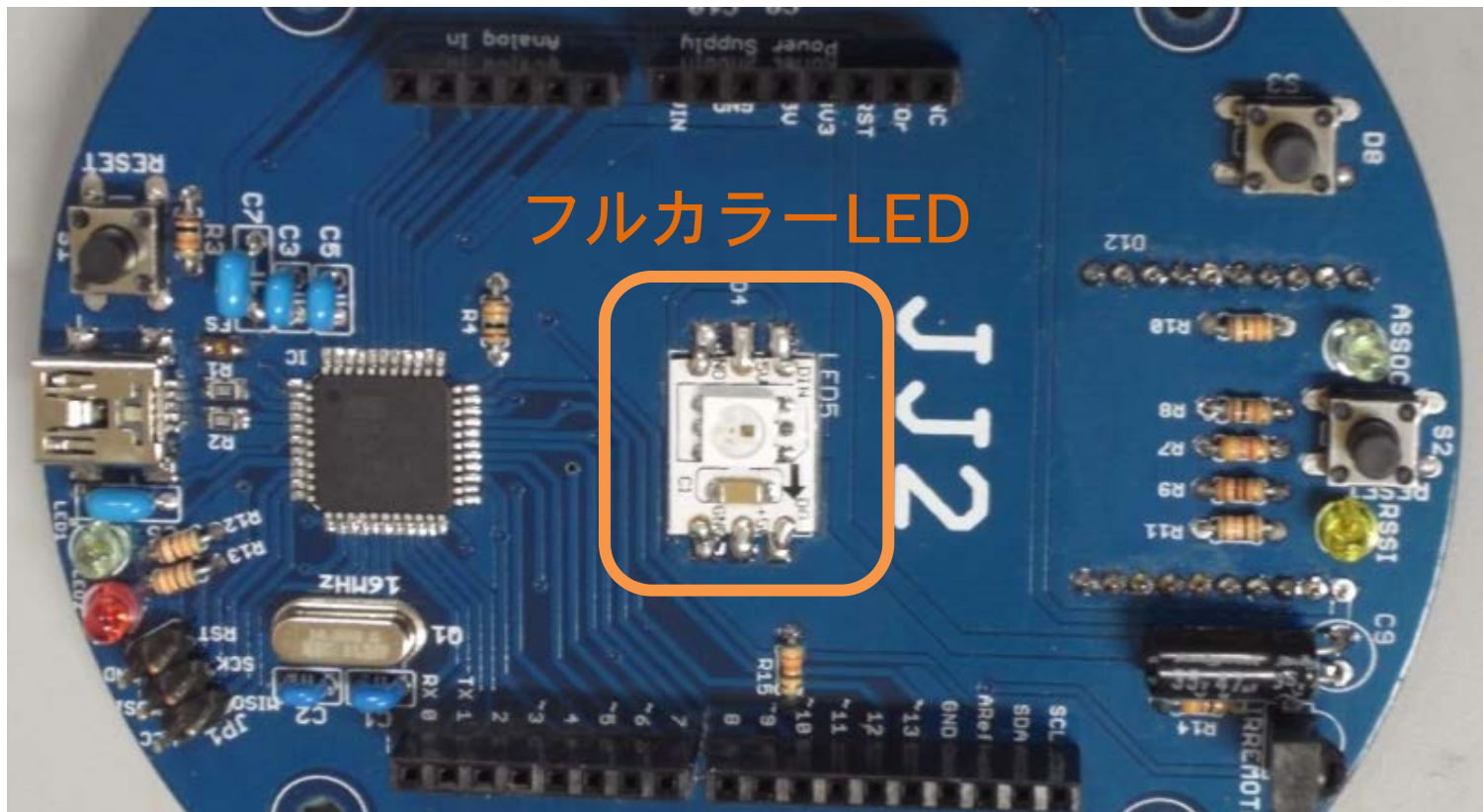
※ 電源LEDが点灯しない場合，すみやかに電源を切り原因を調査する。



電源LED（黄色 3 V3，赤 5 V）

フルカラーLEDが点灯（テストプログラム）

5. 電源が供給されると、（あらかじめ書き込まれた）テストプログラム（フルカラーLEDの点灯）が実行される



マイコンボードの電源の切り方

- ・電源スイッチをBATTERY側に切り替え, または
- ・USBケーブルを抜く

※ ただし, プログラム書き込み中に電源を切らない.



電源スイッチ (BATTERY側にスライド)

補足：手動によるUSBドライバのインストール

マイコンが認識されない！

(マイコンボード接続時にCOMポートが認識されない)

次の手順に従い、手動でUSBドライバをインストール：

1. コントロールパネルよりデバイスマネージャを起動
2. ポート（COMとLPT）を開く
3. 不明なデバイスをダブルクリック
4. ドライバの更新
5. コンピュータを参照してドライバを検索
6. 参照， C:\¥arduino-1.0.5-r2を選択し次へ
7. ドライバがインストールされる
8. デバイスマネージャのポート（COMとLPT）を開き，
COMxxが出来ていることを確認

Arduinoの起動

1. デスクトップ上のArduinoアイコンをダブルクリック



Arduinoのアイコン

```
Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

Done Saving.
Binary sketch size: 1026 bytes (of a 30720 byte maximum)

18 Arduino Duemilanove w/ ATmega328 on COM28
```

Arduinoの設定変更（はじめて起動したときのみ）

2. マイコンボードの選択

メニューのツール
→マイコンボード
Arduino Leonardo
を選択

3. シリアルポートの選択

メニューのツール
→シリアルポート
COMxx （xxは数字）
を選択

4. プログラム書き込み状態の表示

メニュー→環境設定
より詳細な情報を表示する
「コンパイル」「書き込み」
にチェック

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

LEDを点滅させる

5 LEDを13番ピンに接続して，ファイル→スケッチの例
→01:digital → Blink を開き，書き込み



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for "Verify", "Upload", "New", "Open", "Save", and "Find". The sketch name "Blink" is displayed in the top bar. The main text area contains the following code:

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly.  
  
This example code is in the public domain.  
*/  
  
void setup() {  
  // initialize the digital pin as an output.
```

LEDを点滅させる

6 delay関数を書き換えて，LEDの点滅時間を変える。
※ delay は ms（ミリ秒）で設定

```
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);          // wait for a second  
}
```

Done Saving.

Binary sketch size: 1026 bytes (of a 30720 byte maximum)

18

Arduino Duemilanove w/ ATmega328 on COM28

7 マイコンボードについて

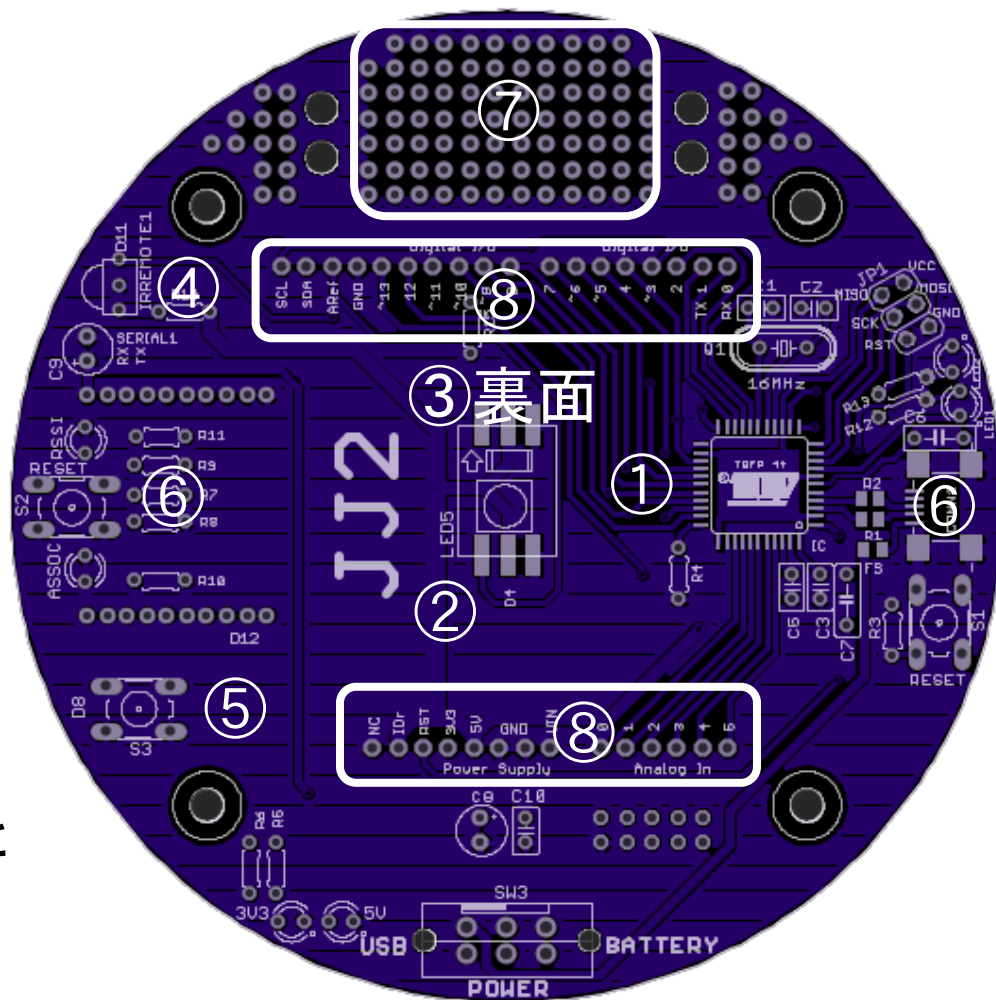
マイコンボード

JJ にできること

- ① マイコンのプログラム
- ② フルカラーLEDの点灯
- ③ スピーカを鳴らす
- ④ リモコンによる操作
- ⑤ プログラム起動スイッチ
- ⑥ 有線・無線通信
- ⑦ センサ増設など拡張エリア

Arduino Leonardo互換ボードとして使用可能

- ⑧ Arduino互換ピンソケット



マイコンボードのマイコン

JJ のマイコン

ATmega 32u4 (AVR社)

主な機能：

8ビットRISCマイコン
クロック 16 MHz

32kB FlashROM

2.5kB RAM

1kB EEPROM

タイマ8ビット×1

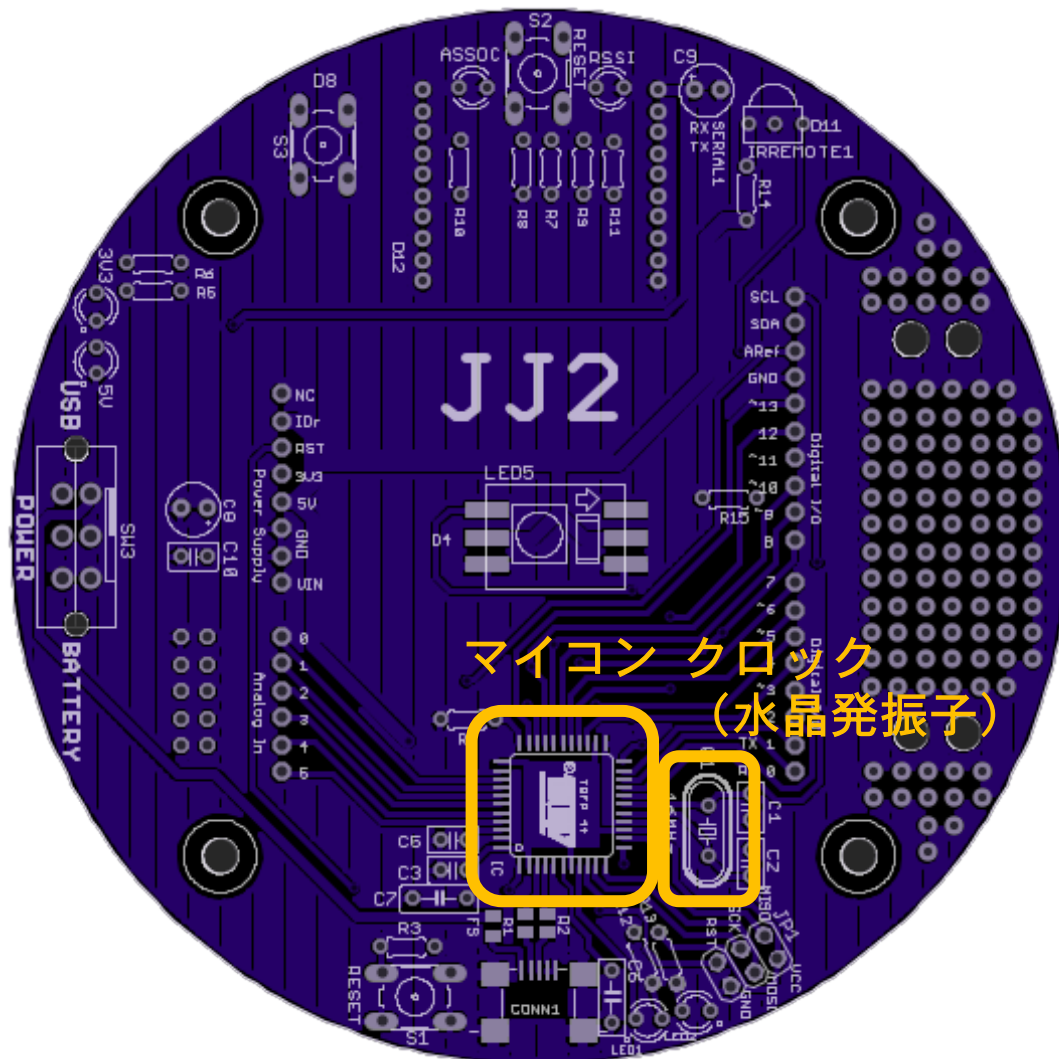
タイマ16ビット×2

10ビットAD変換器×12

シリアル通信USARTx1

I2C通信x1

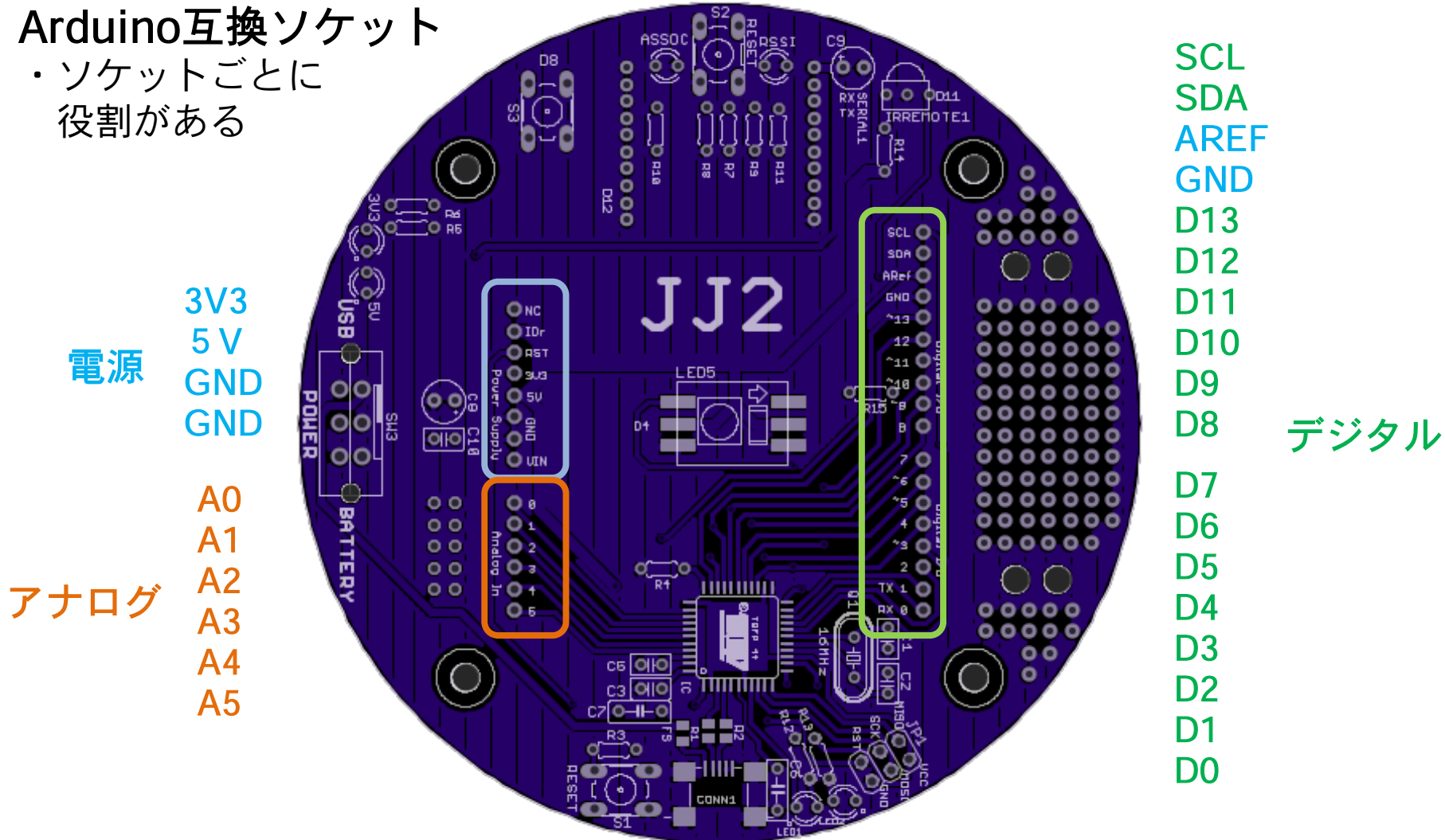
SPI通信x1



マイコンボードのArduino互換ソケット

Arduino互換ソケット

- ・ソケットごとに役割がある



付録 テープLEDの光らせ方

1.1 RGB表色系

光の三原色

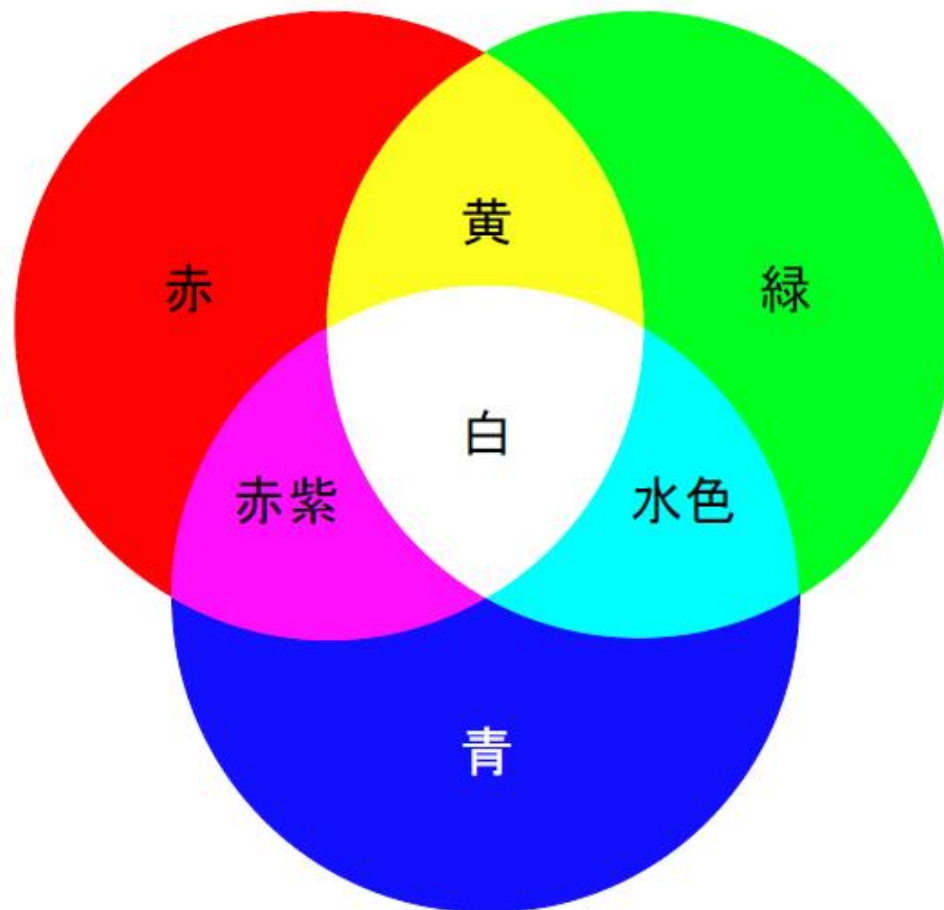
- ・ 赤(Red)
 - ・ 緑(Green)
 - ・ 青(Blue)
- の重ね合わせ

CRGB(R, G, B);

R値 : 0 - 255 赤の明るさ

G値 : 0 - 255 緑の明るさ

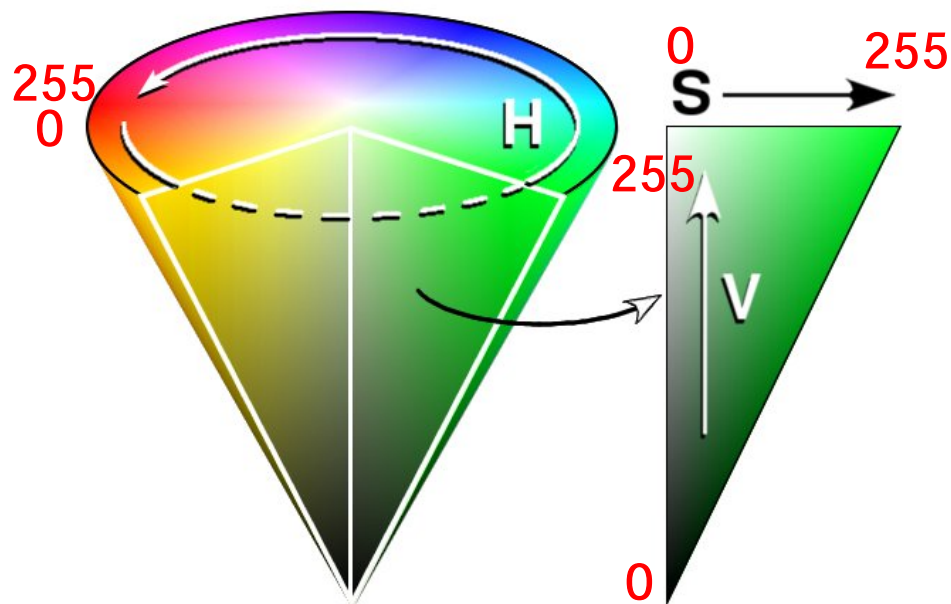
B値 : 0 - 255 青の明るさ



1.2 HSV表色系

HSV表色系

- 色相(H: Hue)
色の種類
- 彩度(S: Saturation)
色の鮮やかさ
- 明度(V: Value, Brightness)
色の明るさ

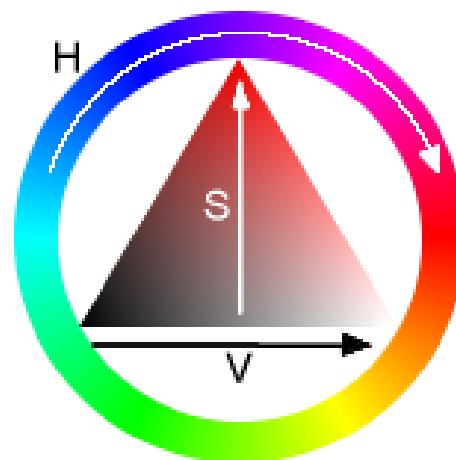


CHSV(h, s, v);

h値 : 0 - 255

s値 : 0 - 255

v値 : 0 - 255



2 テープLEDを光らせる

2.1 スケッチの基本

```
#include "FastLED.h"

const int LEDPin = 4;  // 4番ピン

CRGB led[1];  // LEDの構造体

void setup() {
  delay(2000);
  FastLED.setBrightness(32); // LEDの最大の明るさ(適宜変更)
  FastLED.addLeds<WS2811, LEDPin, GRB>(led, 1);
}
```

変更しない

```
void loop() {
  (LEDの光らせ方を書く)
}
```

2.1 スケッチの基本

LED表示の基本

```
void loop() {  
  <LEDに値をセット>  
  FastLED.show(); //LEDの表示更新  
  delay(N); // LEDの表示時間（ミリ秒）  
}
```

<LEDに値をセット>

・ 赤, 緑, 青, それぞれに輝度値を入力

```
led[0].r = 255; // 0--255
```

```
led[0].g = 255; // 0--255
```

```
led[0].b = 255; // 0--255
```

・ 赤, 緑, 青, をまとめて入力

```
led[0] = CRGB(r, g, b);
```

・ 色相, 彩度, 明度をまとめて入力

```
led[0] = CHSV(h, s, v);
```


スケッチ例1

(1) テープLED1個, 赤を点灯

```
void loop() {  
  led[0].r = 255;  
  led[0].g = 0;  
  led[0].b = 0;  
  FastLED.show();  
  delay(10);  
}
```

(2) テープLED1個, 赤を点灯 ((1)の別の書き方)

```
void loop() {  
  led[0] = CRGB(255, 0, 0);  
  FastLED.show();  
  delay(10);  
}
```

(3) テープLED1個, 色相 (H) を変化

```
void loop() {  
  led[0] = CHSV(0, 255, 255); // red  
  FastLED.show();  
  delay(10);  
}
```

その他の色 :

```
led[0] = CHSV(32, 255, 255); // orange  
led[0] = CHSV(64, 255, 255); // yellow  
led[0] = CHSV(96, 255, 255); // green  
led[0] = CHSV(128, 255, 255); // aqua  
led[0] = CHSV(160, 255, 255); // blue  
led[0] = CHSV(192, 255, 255); // purple  
led[0] = CHSV(224, 255, 255); // pink
```

スケッチ例2

(4) 色相, 明るさを固定し, 彩度を変化

```
void loop() {  
  for (int i=0; i<255; i++) {  
    led[0] = CHSV(32, i, 255);  
    FastLED.show();  
    delay(10);  
  }  
}
```

(5) 色相, 彩度を固定し, 明るさを変化

```
void loop() {  
  for (int i=0; i<255; i++) {  
    led[0] = CHSV(32, 255, i);  
    FastLED.show();  
    delay(10);  
  }  
}
```

(6) LEDを赤に点灯 (N個) (複数個)

```
void loop() {  
  fill_solid(&(led[0]), N, CRGB(255, 0, 0));  
  FastLED.show();  
  delay(10);  
}
```

(7) LEDの輝度を変化 (N個) (複数個)

```
void loop() {  
  for (int i = 0; i < 255; i++) {  
    fill_solid(&(led[0]), N, CRGB(i,i,i));  
    FastLED.show();  
    delay(10);  
  }  
}
```

スケッチ例3

(8) LEDすべての色相を変化 (N個) (複数個)

```
void loop() {  
  for (int i = 0; i < 255; i++) {  
    fill_solid(&(led[0]), N, CHSV(i, 255, 255));  
    FastLED.show();  
    delay(10);  
  }  
}
```

(9) LEDの彩度を変化 (N個) (複数個)

```
void loop() {  
  for (int i = 0; i < 255; i++) {  
    fill_solid(&(led[0]), N, CHSV(0, i, 255));  
    FastLED.show();  
    delay(10);  
  }  
}
```

付録 Arduino のピン割り当て (詳細)

Arduinoピン割り当て

	Arduino Leonardo	Arduino UNO
Digital IO	0～13, 14(MISO), 15(SCK), 16(MOSI), 17(SS), 18～23(A0～A5)	0～13, 14～19(A0～A5)
PWM	3, 5, 6, 9, 10, 11, 13	3, 5, 6, 9, 10, 11
Analog	A0～A5 A6～A11(4, 6, 8, 9, 10, 12)	A0～A5
I2C(Wire)	2, 3 (SDA, SCL)	A4(SDA), A5(SCL)
SPI	ICSP-4, ICSP-1, ICSP-3 (MOSI, MISO, SCK)	11, 12, 13, 10 (MOSI, MISO, SCK, SS) ICSP-4, ICSP-1, ICSP-3
Interrupt	3 (INT0), 2(INT1), 0 (INT2), 1(INT3)	2(INT0), 3(INT1)
Serial	0,1 (Rx, Tx: Serial1)	0, 1(Rx, Tx: Serial)
Software Serial	8, 9, 10, 11, 14(MISO), 15(SCK), 16(MOSI) (Rx), 0～23(Tx)	0～19(Rx), 0～19(Tx)

ArduinoのTimer

	Arduino Leonardo	Description
Timer 0 (8bit)	3, 11	delay, millis, micros (Timer0), analogWrite
Timer 1, 3 (16bit)	5, 9, 10(Timer1/Timer3) 11(Timer1)	Servo L(Timer1), pin. 9 Servo R(Timer1), pin. 5 tone(Timer3), pin.10 IRremote, pin.11 analogWrite
Timer 4 (10bit)	6, 13(Timer4), 10(Timer1)	IRremote Ir.Send, pin.13 (IRremote(Timer2), pin.3 UNO) analogWrite