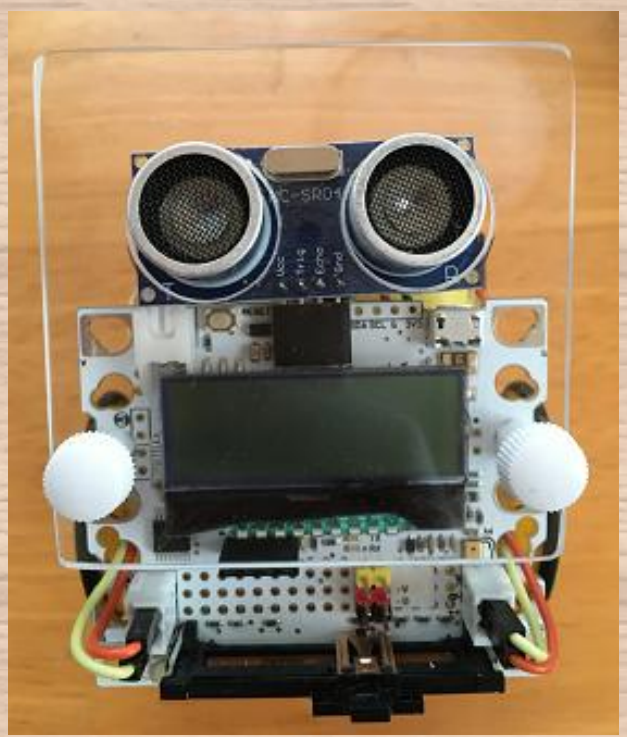
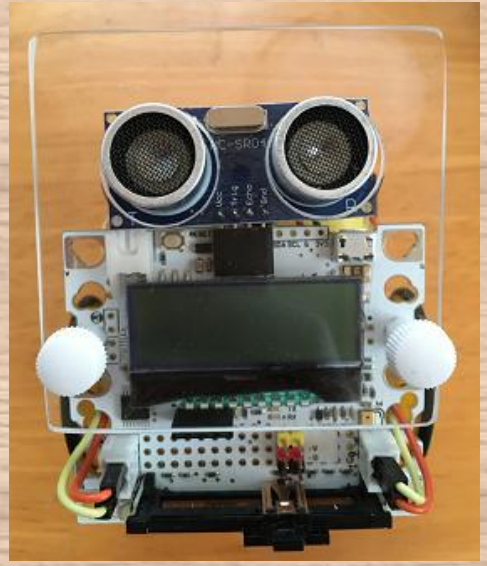
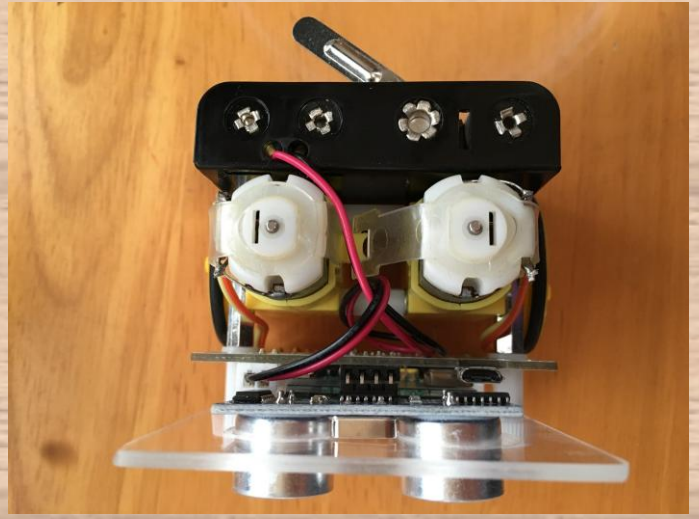
ロボットプログラミング



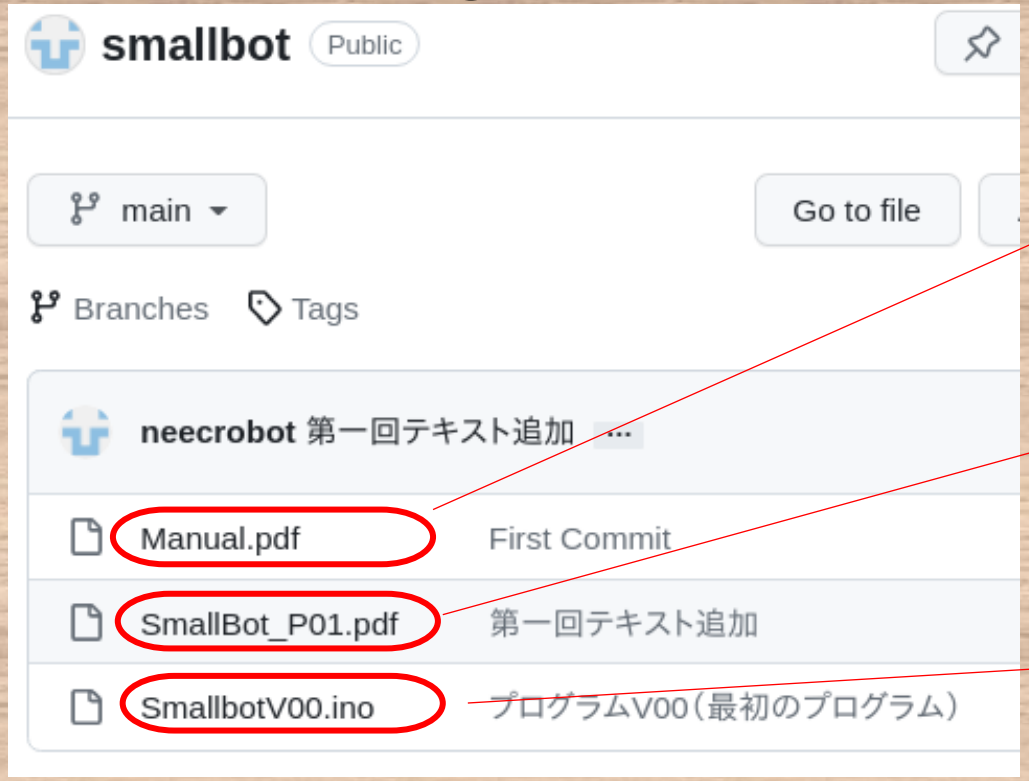
SmallBotを動かそう

SmallBotについて





マニュアルやプログラム類は githubに置いてあります



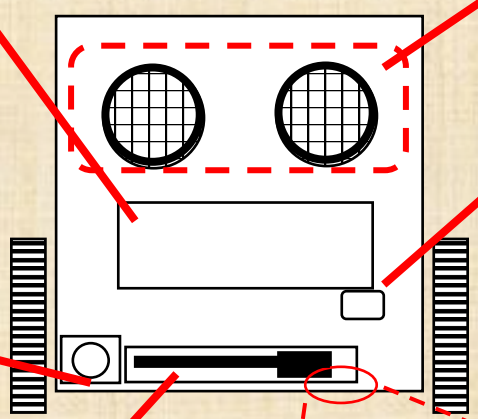
Manual.pdf
ハードウェアマニュアル

SmallBot_P01.pdf
第一回テキスト

SmallbotV00.ino
サンプルプログラム

https://github.com/neecrobot/smallbot

SmallBotの主要部品配置



キャラクタLCD

超音波距離センサ

右モーター

マイク

押しボタンスイッチ

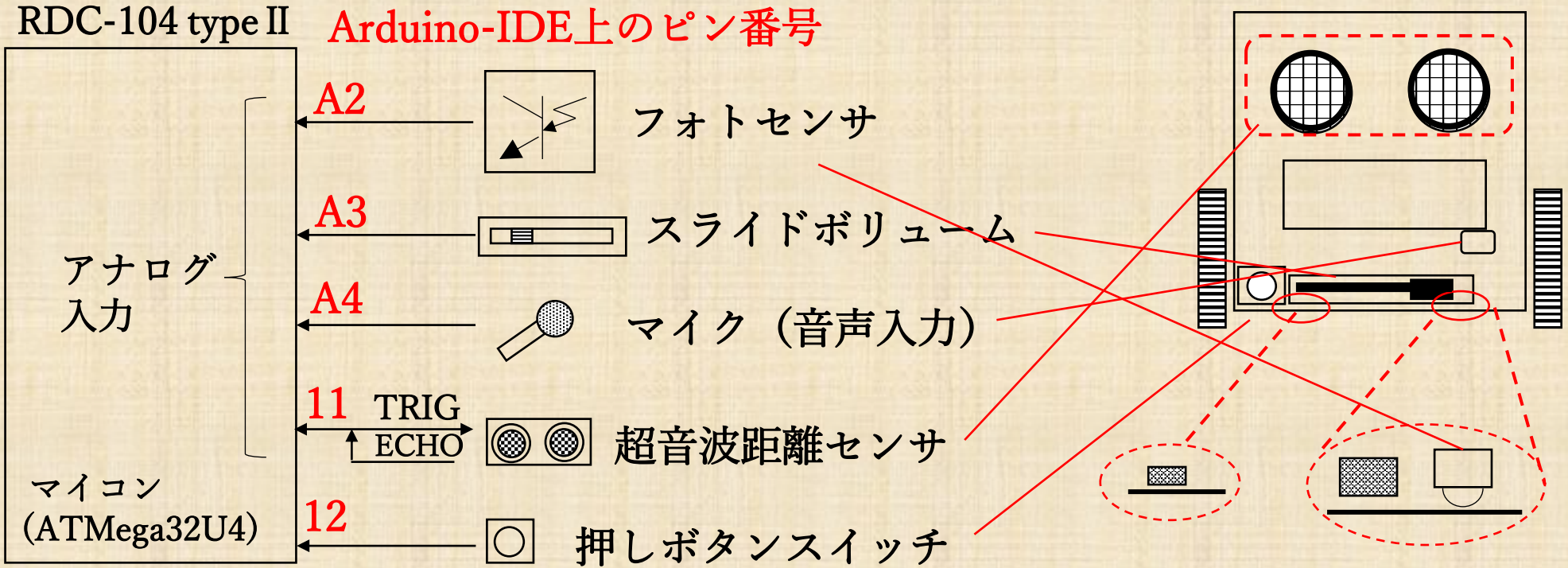
左モーター

スライドボリューム

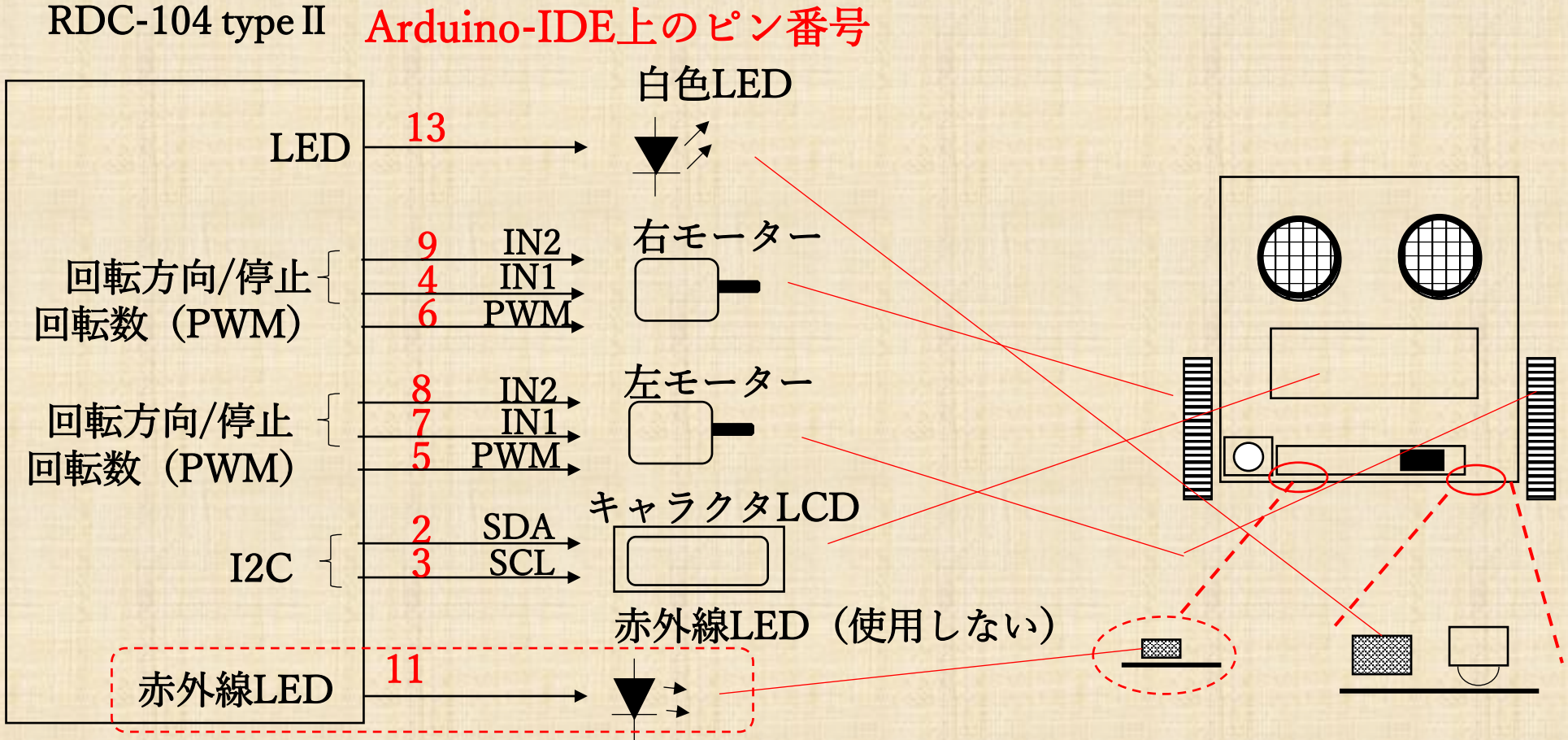
白色LED

フォトセンサー

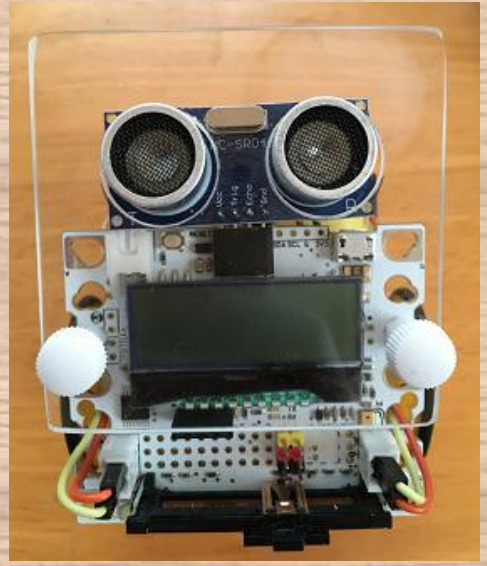
入力信号の接続（参考）

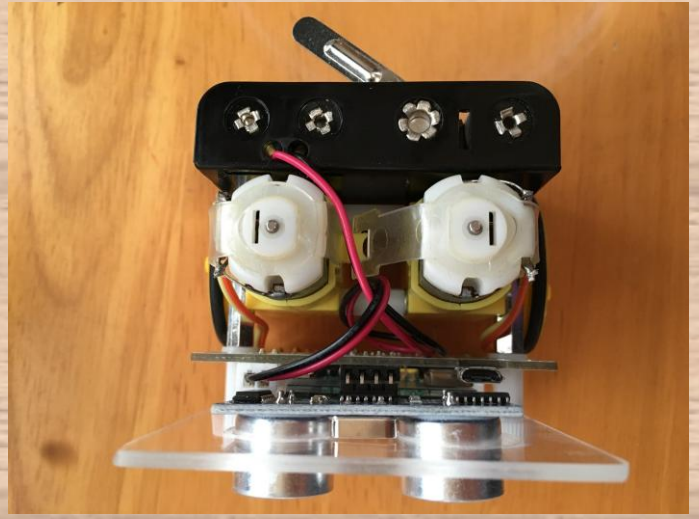

RDC-104 type II
Arduino-IDE上のピン番号
A2
フォトセンサ
A3
スライドボリューム
アナログ
入力
A4
マイク（音声入力）
11
TRIG
ECHO
超音波距離センサ
マイコン
(ATMega32U4)
12
押しボタンスイッチ

出力信号の接続（参考）

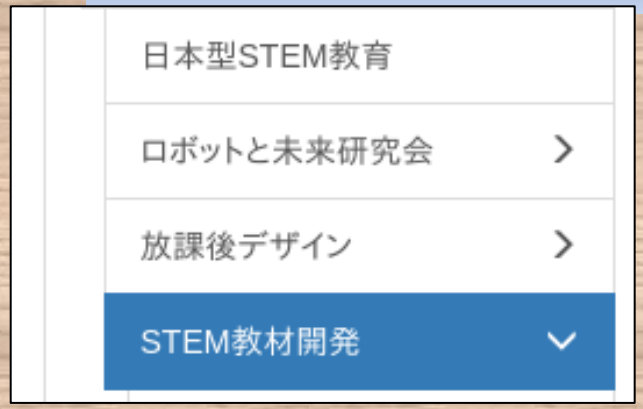

RDC-104 type II
Arduino-IDE上のピン番号
白色LED
LED
13
右モーター
回転方向/停止
回転数（PWM）
9 IN2
4 IN1
6 PWM
左モーター
回転方向/停止
回転数（PWM）
8 IN2
7 IN1
5 PWM
キャラクタLCD
I2C
2 SDA
3 SCL
赤外線LED（使用しない）
赤外線LED
11

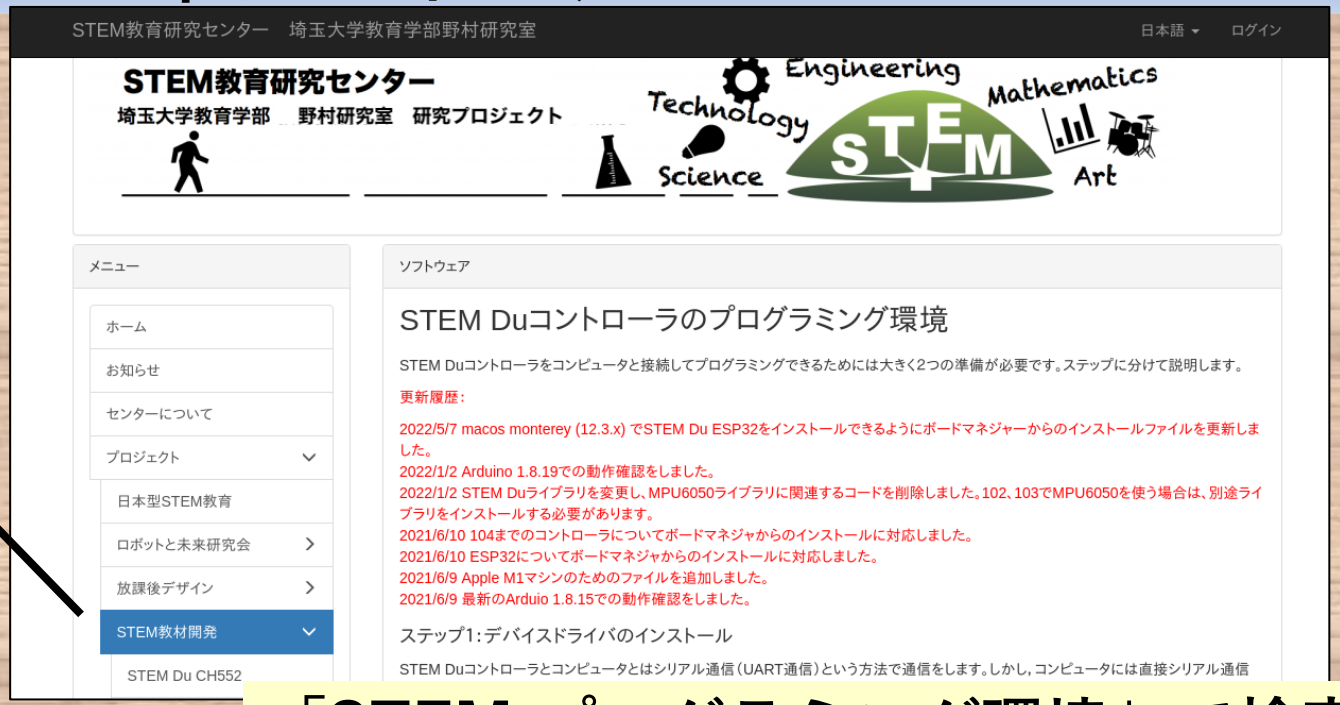
SmallBot搭載マイコン用(RDC-104)の開発環境インストール





SmallBot搭載マイコン用(RDC-104)の開発環境インストール

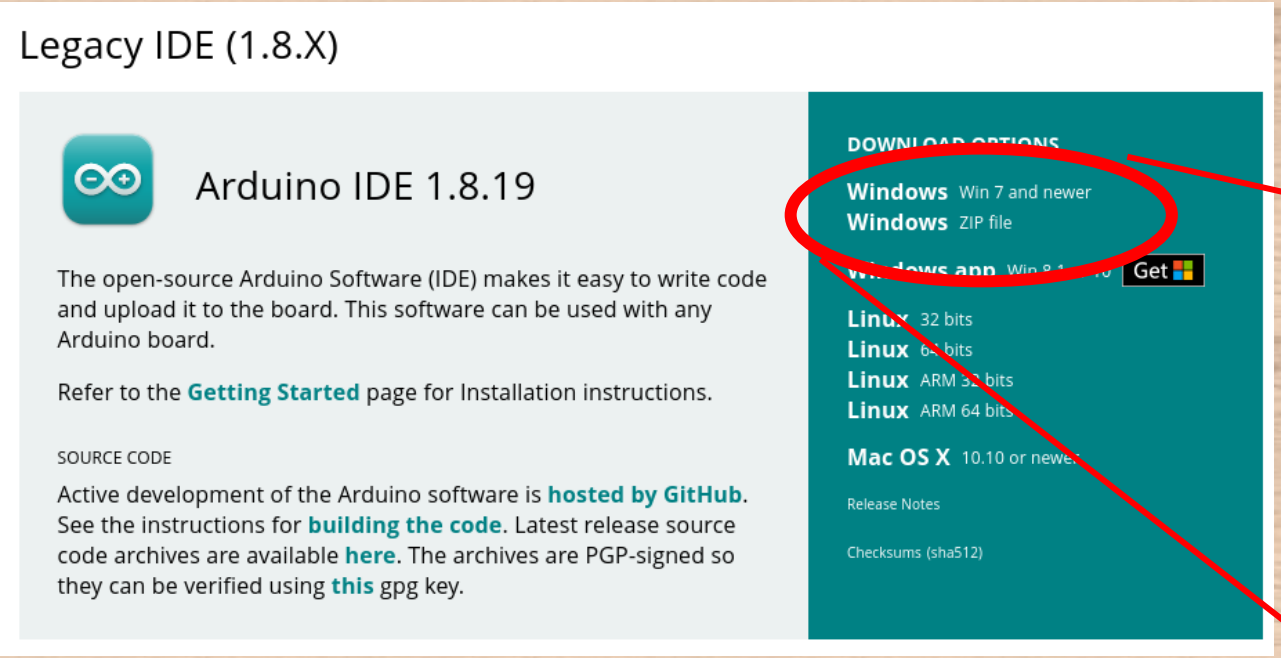


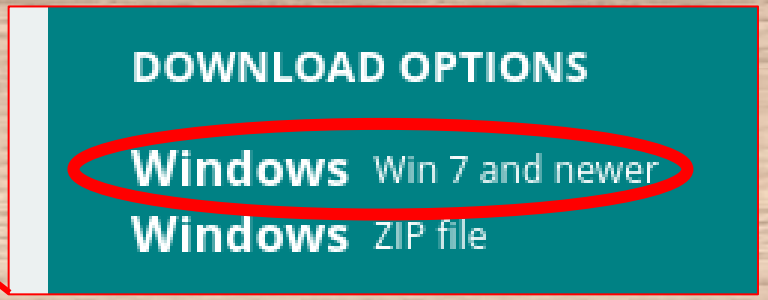


「STEM プログラミング環境」で検索
=>埼玉大学のSTEM教育研究センター

http://neo.stem-edulab.org/page_20200820030156/page_20201016230943

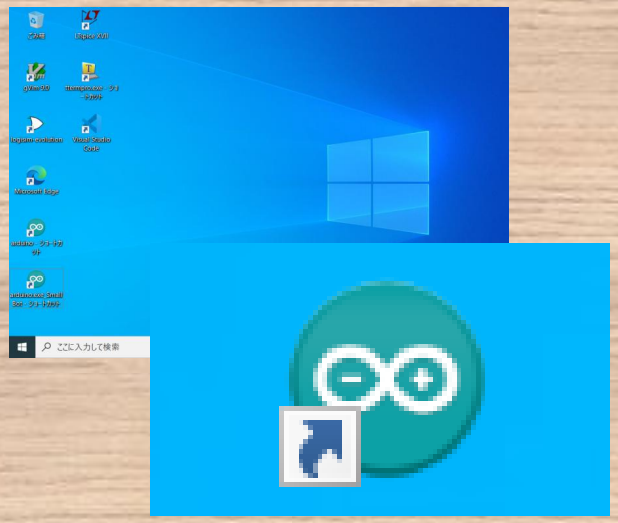
Arduino 1.8.19をインストール

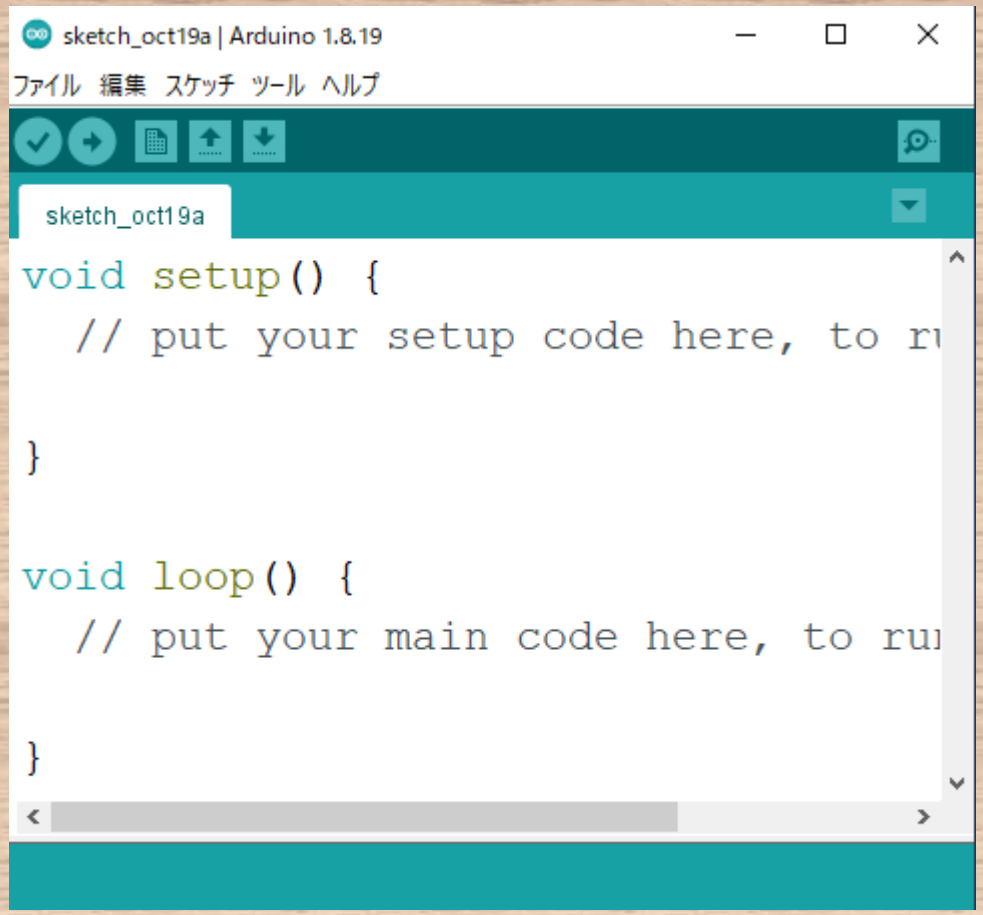




https://www.arduino.cc/en/software

Arduino IDEの起動



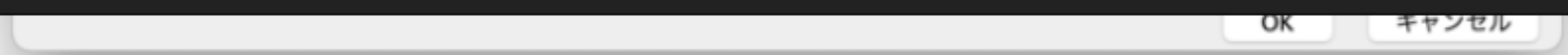


既存のプログラムが開いたら

ファイル=>新規ファイル

ボードマネージャのURLをコピー

STEM教育研究センター　埼玉大学教育学部野村研究室



２．上の図のように「追加のボードマネジャーのURL」の右の欄に、使用するSTEM Duマイコンボードのバージョンに合わせて、次のいずれかのURLをコピー＆ペーストします。

【STEM Du RDC-ESP32コントローラの場合】

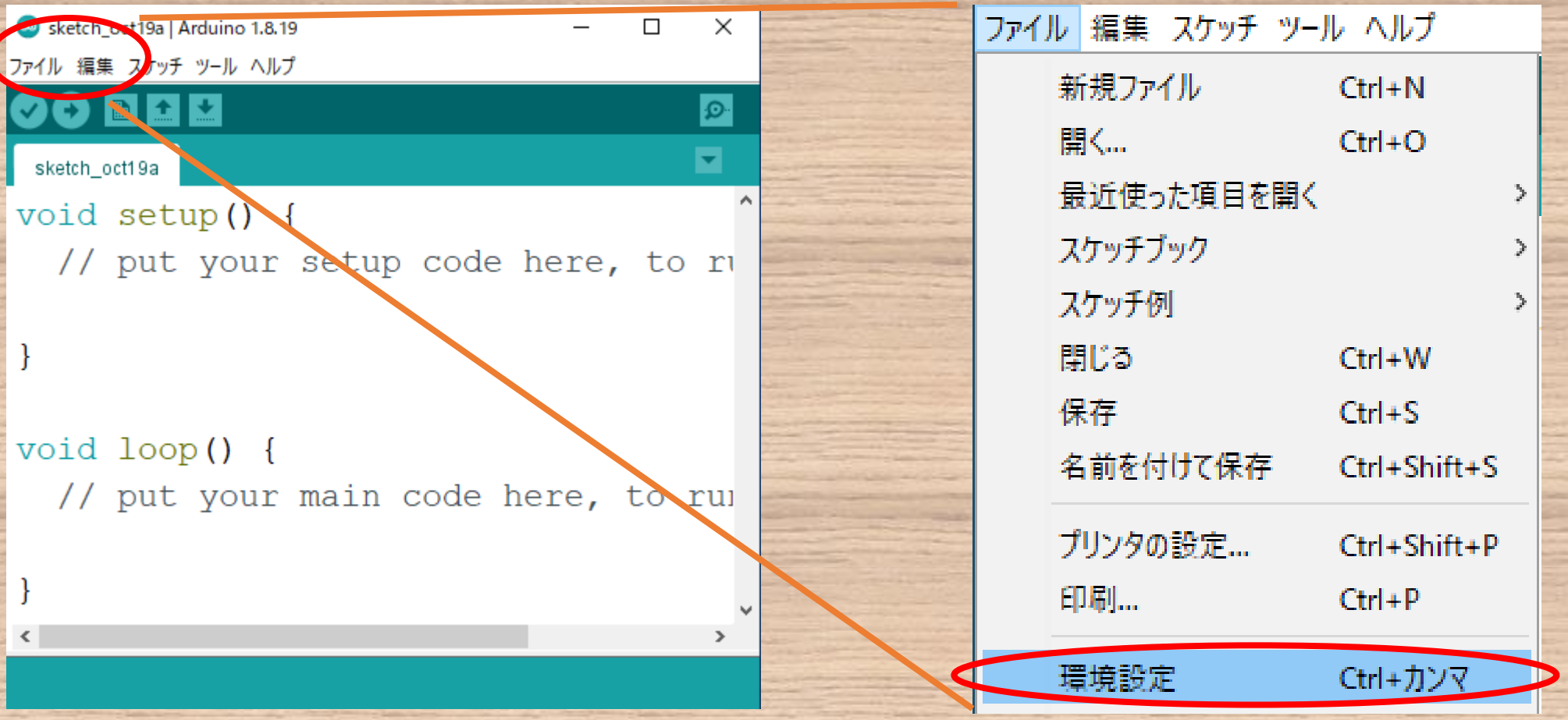
https://github.com/tairo/STEMDu_ArduinoIDE/releases/download/v1.0/package_stemdu-esp32_index.json

【STEM Du RDC-104までのコントローラの場合】

https://github.com/tairo/STEMDu_ArduinoIDE/releases/download/v1.0/package_stemdu-avr_index.json

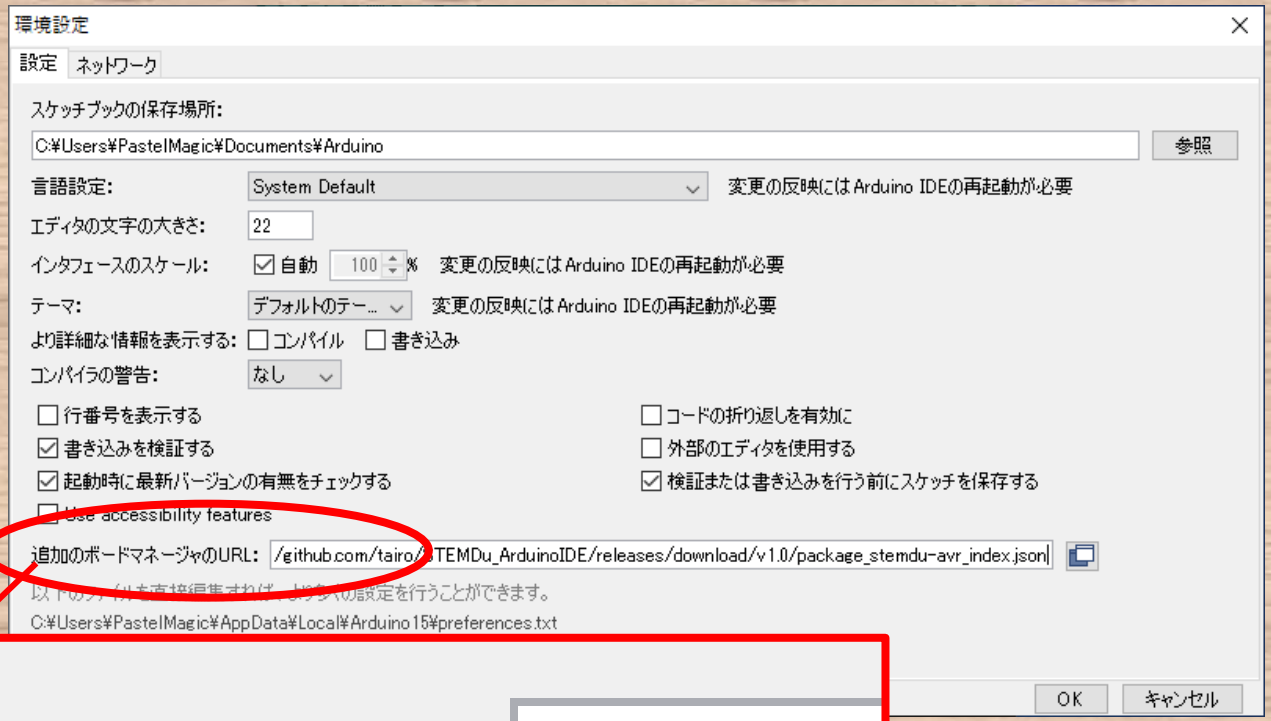
https://github.com/tairo/STEMDu_ArduinoIDE/releases/download/v1.0/package_stemdu-avr_index.json

ボード（RDC-104）の追加



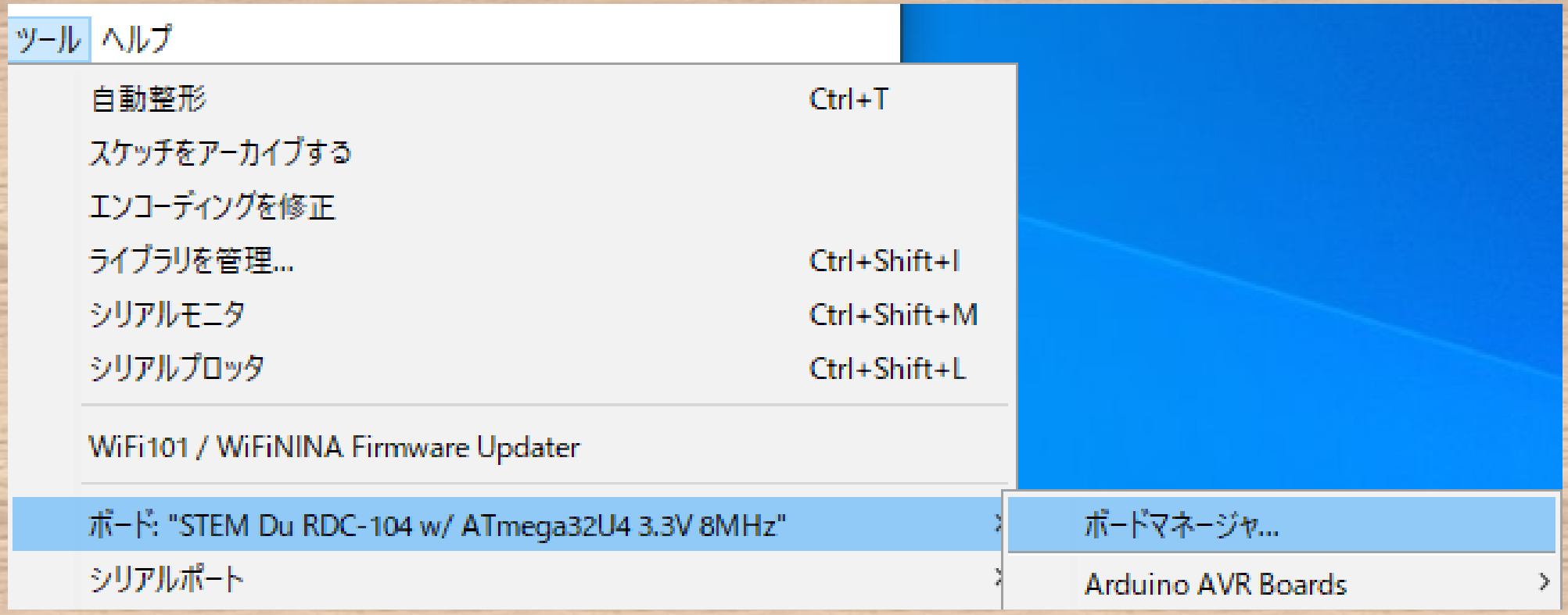
ボードマネージャのURL追加

コピーしたURLをペースト



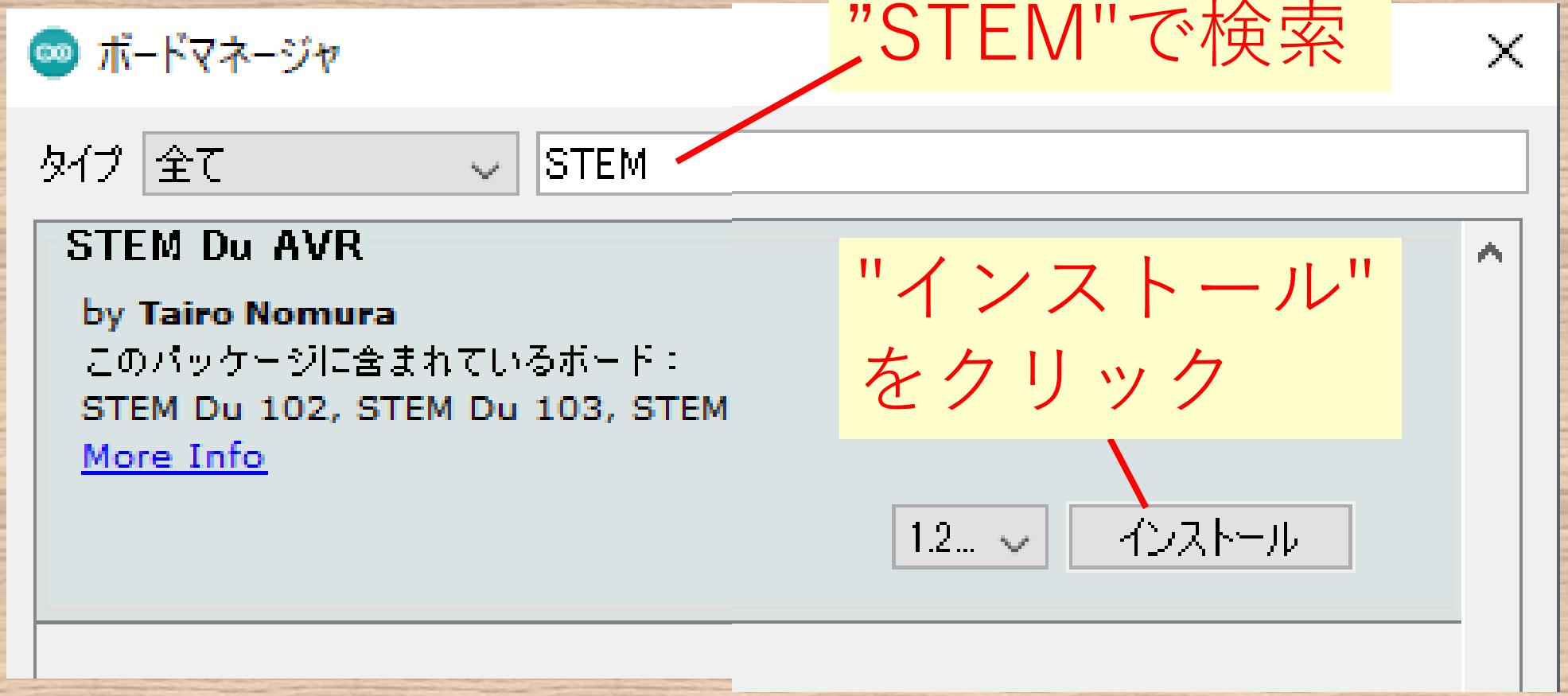
追加のボードマネージャのURL: /github.c

ボードマネージャを起動

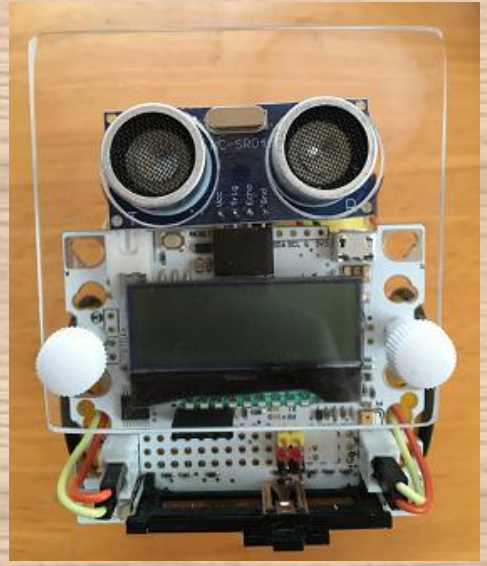


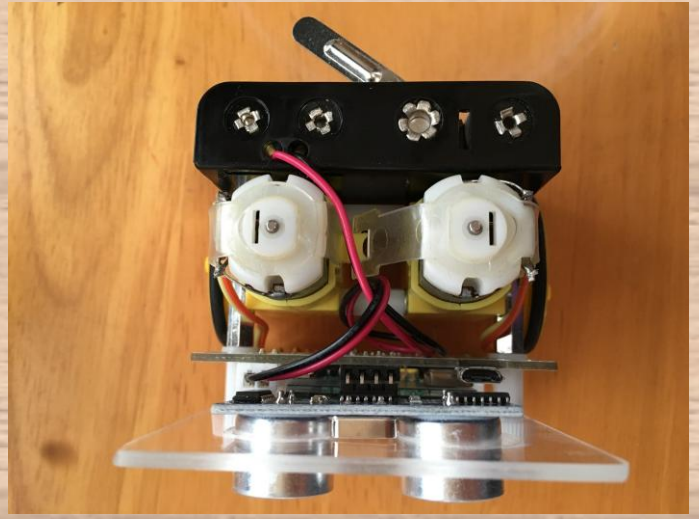
ツール ＝＞ ボード ＝＞ ボードマネージャ

STEM Du AVRをインストール

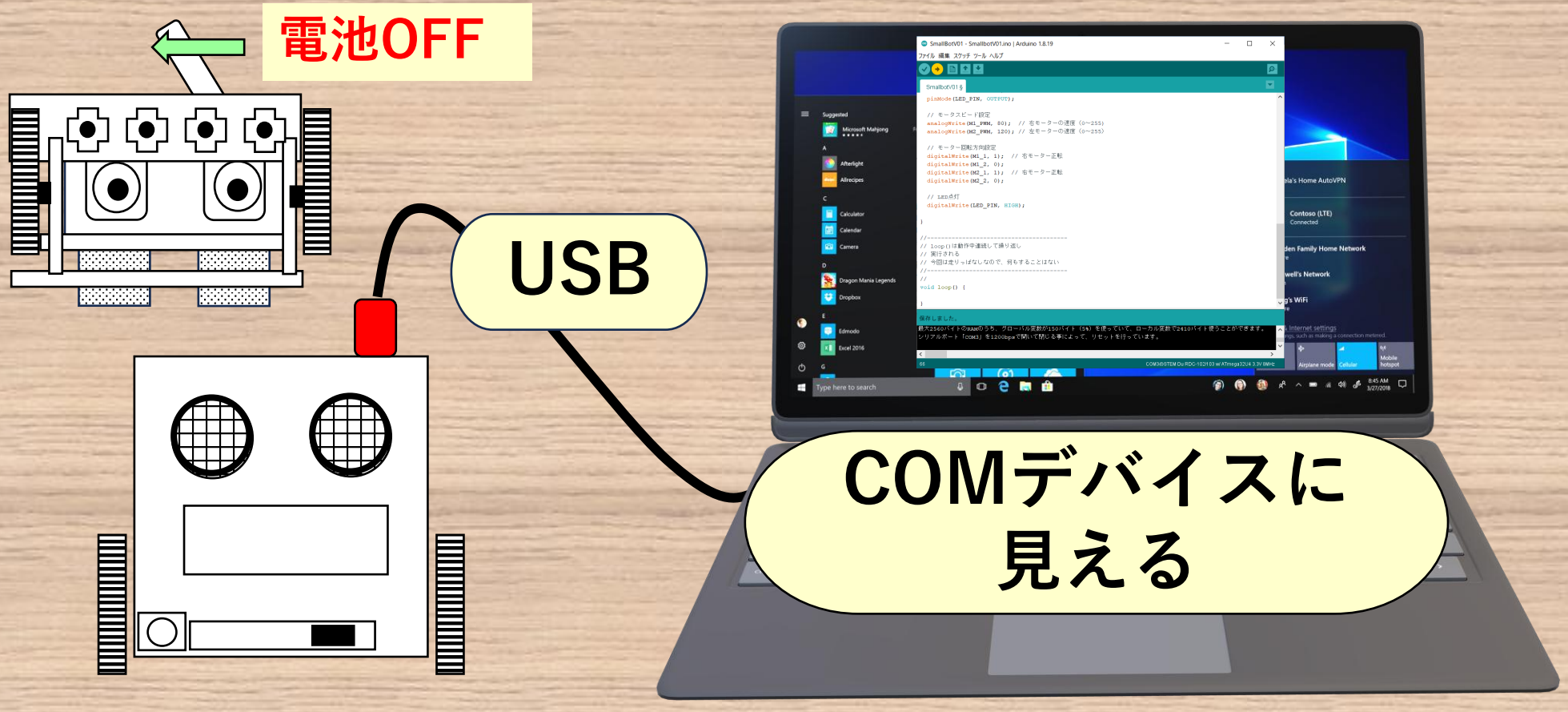


書き込んで動かすまでの手順

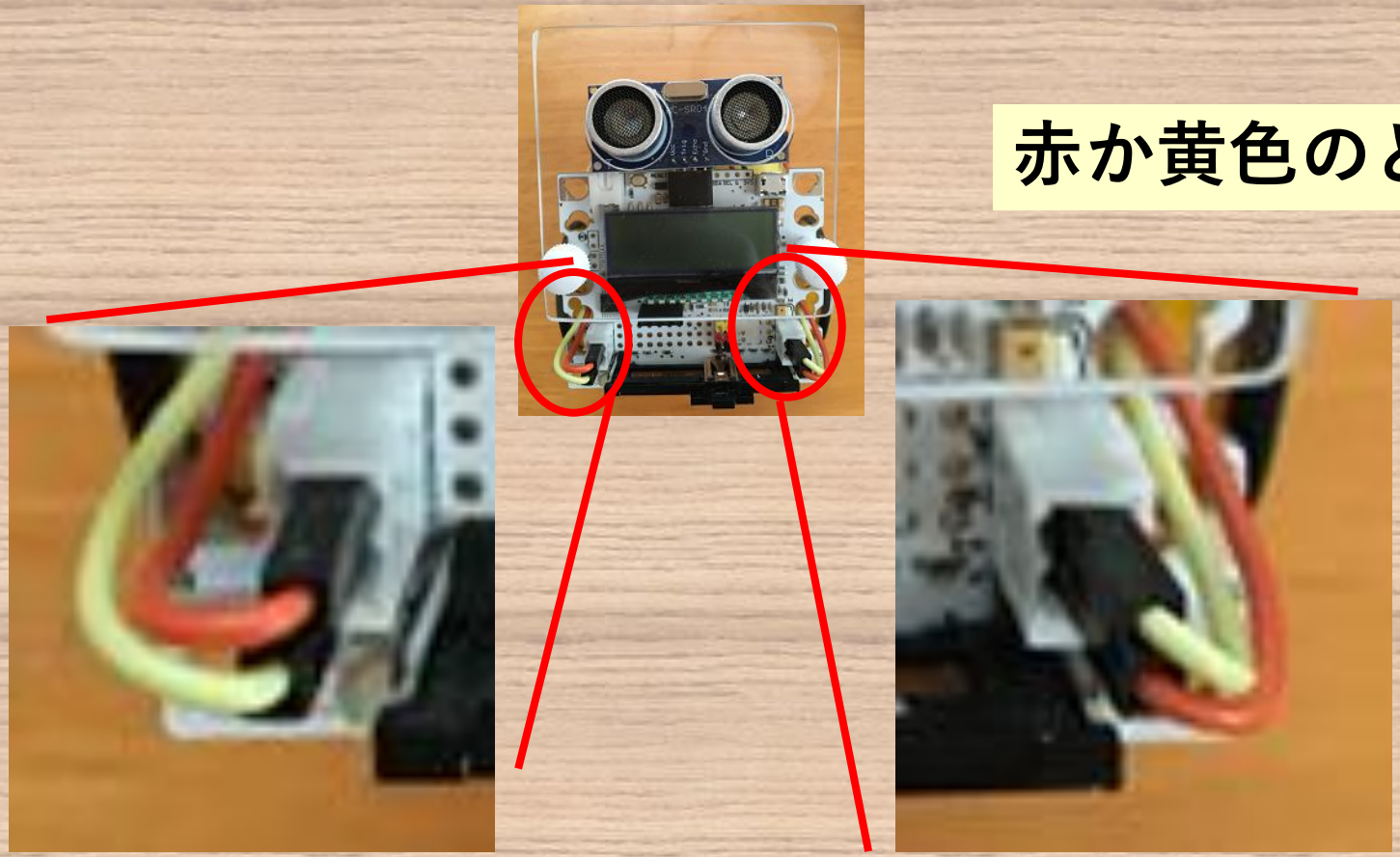




PCと接続

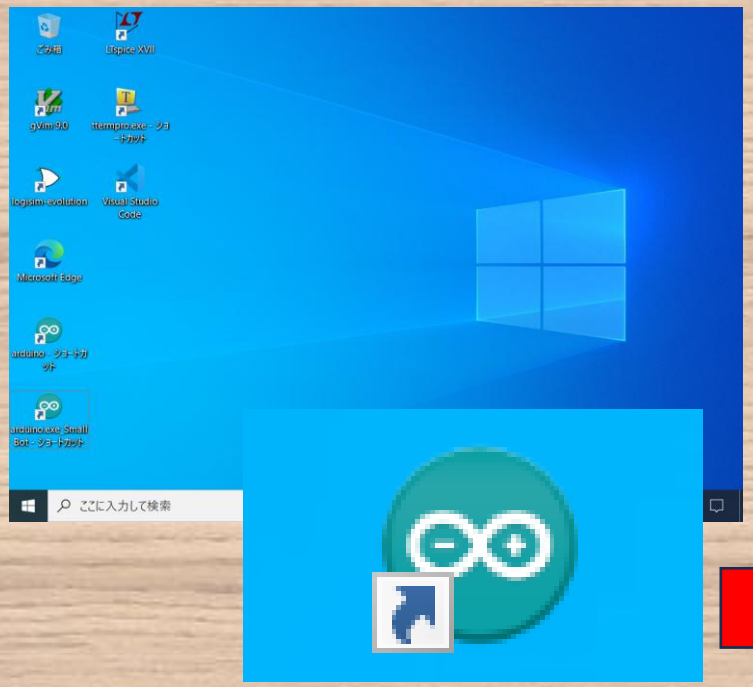


USB電源で動いてしまう時



赤か黄色のどちらかを抜く

Arduino IDEの起動



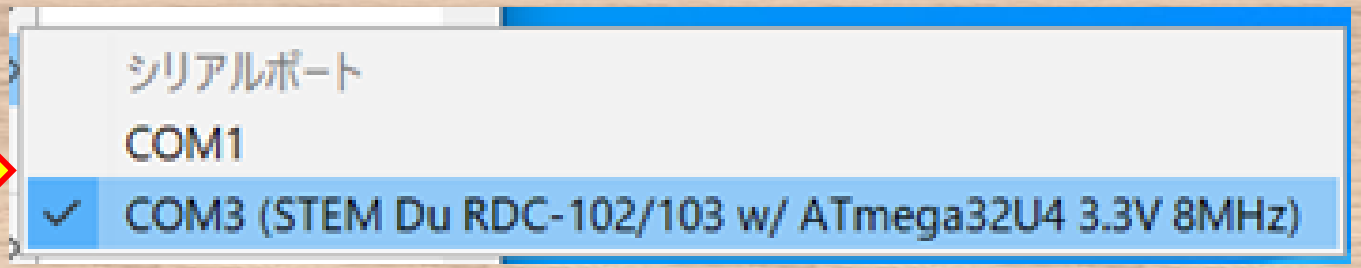
ターゲットの選択

ツール =>ボード =>Arduino AVR Boards
=>STEM Du RDC-104.... を選択

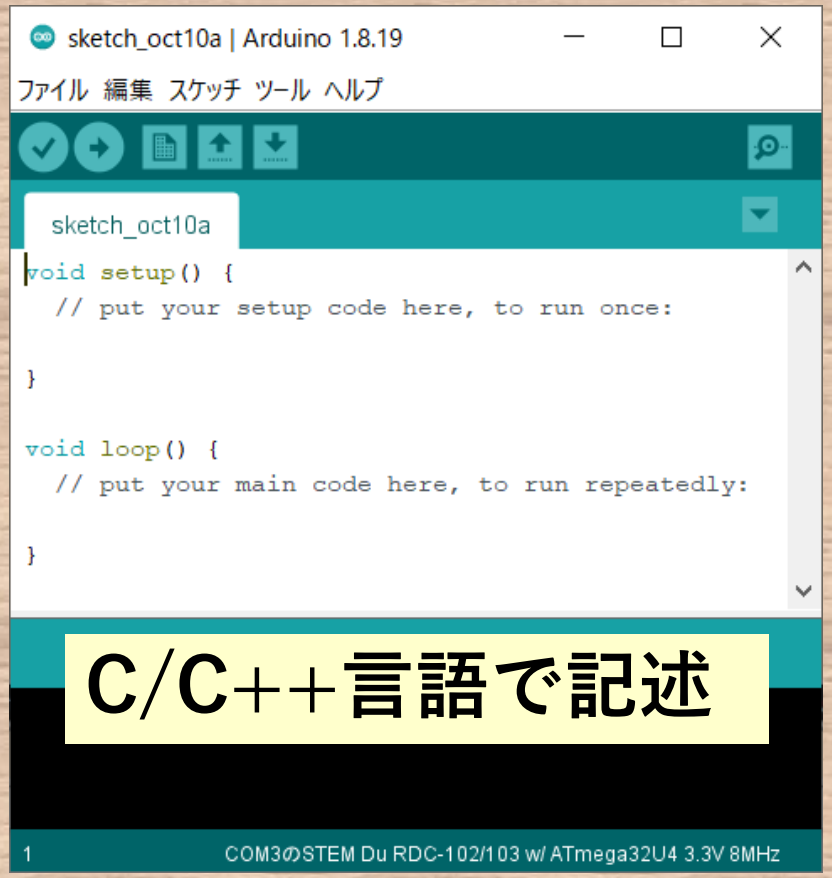
COM（シリアル）ポートの選択

COM(シリアル）ポートの番号はPC環境によって異なる

ツール
=>シリアルポート
=>STEM Du RDC-102/103 w/……



プログラムの記述
(setup()やloop()を消去しないこと）




```
void setup() {
```

起動時最初に
一回だけ実行される

```
}

void loop() {
```

繰り返し実行される

```
}
```

とりあえずLED点滅

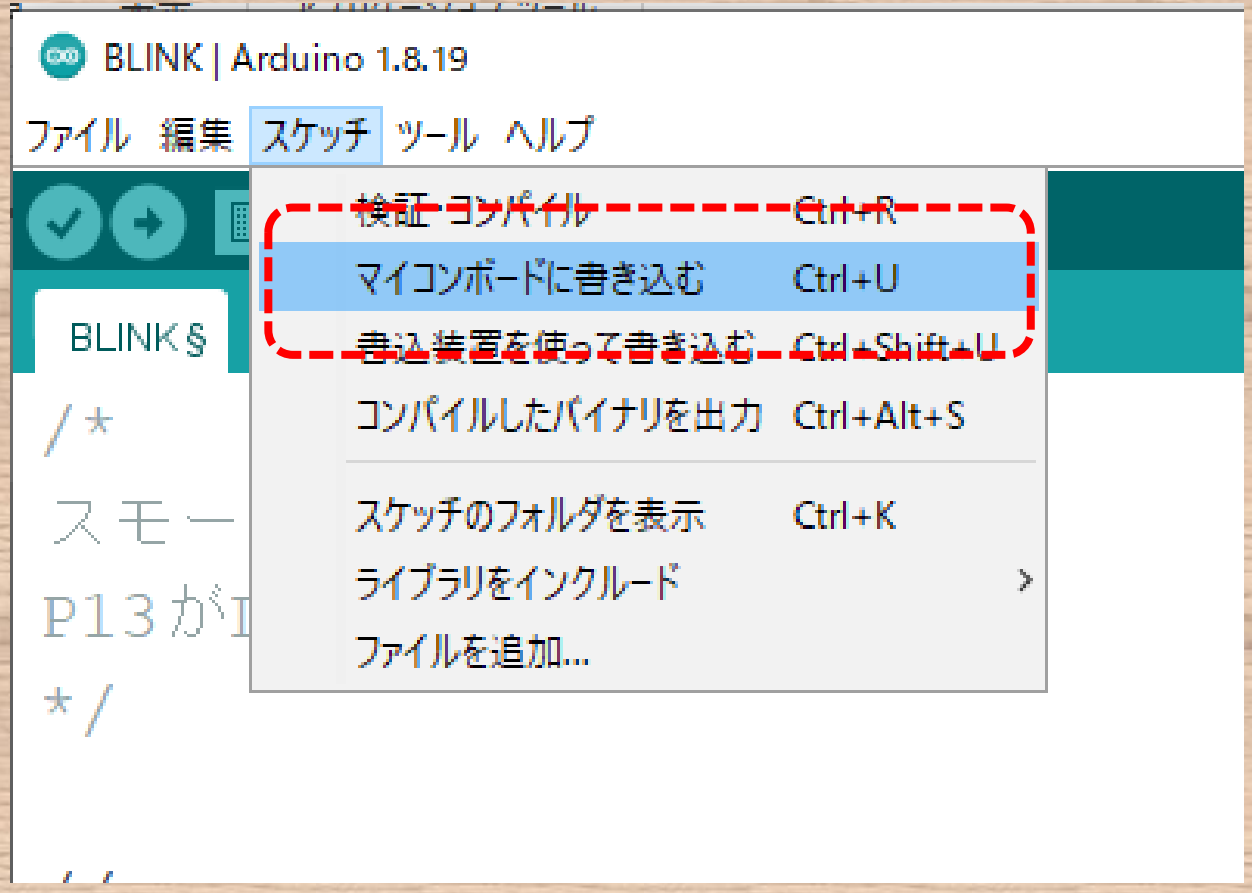

```
void setup() {
 pinMode(13, OUTPUT);
}

void loop() {
 digitalWrite(13,1); // ON
 delay(500);  // 500msウェイト
 digitalWrite(13,0); // OFF
 delay(500); // 500msウェイト
}
```

"//"以下はコメント（入力不要）

プログラム本体はすべて半角文字（スペースも半角！）

ビルド＆書き込み＆実行

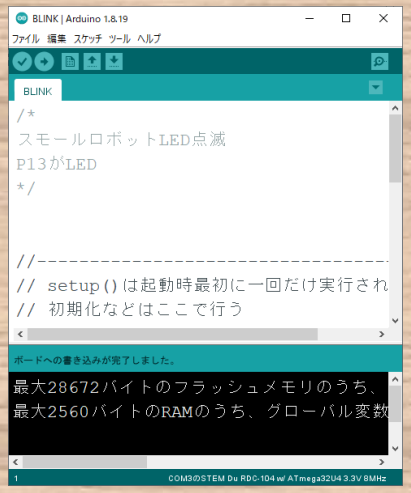


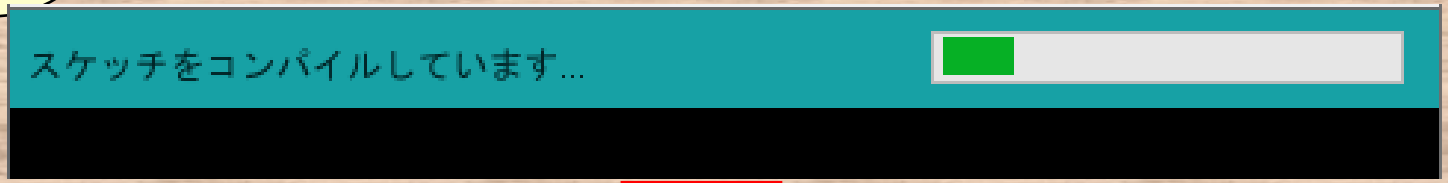
接続したSmallBotに書き込みを行う

ビルドも自動的に行われる

書き込み実行

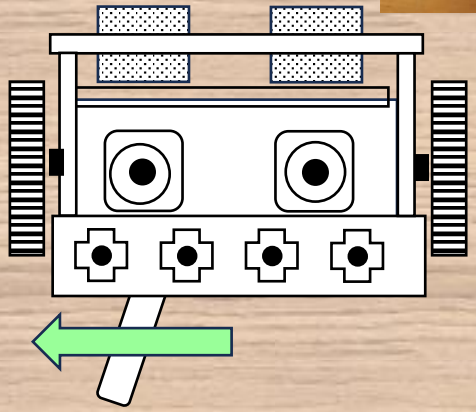
ビルド中

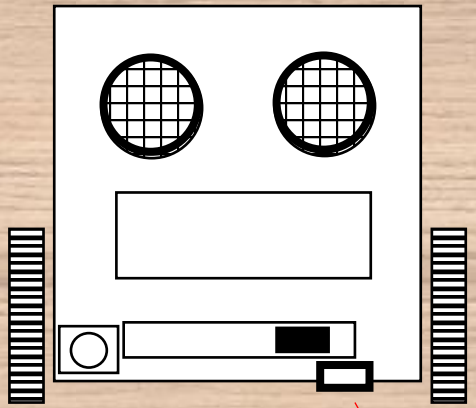




書込完了

電池から電源を供給





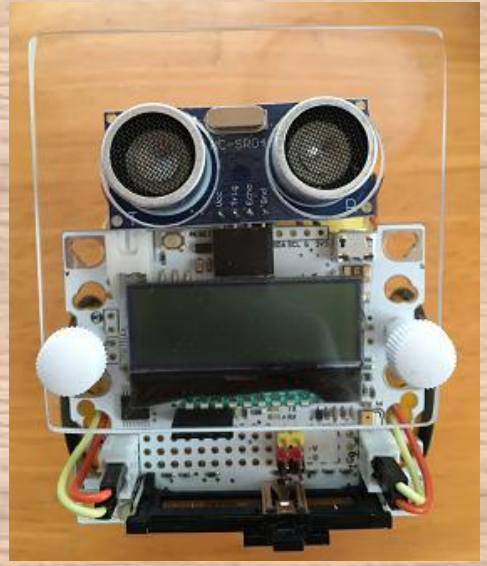
基板上のLEDが1秒周期で点滅する

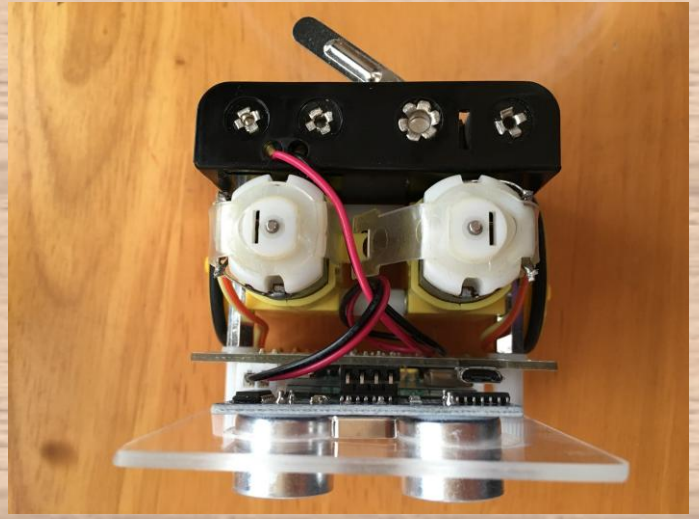
点滅の周期などを書き換えて動作を確認してみよう

電池ON

白色LEDが点滅

モーターを動かそう





最初のロボットプログラム(V00.ino)

(setup()の中に記述)

```
void setup() {
  // モータ関係の出力ピン
  pinMode(4, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(5, OUTPUT);
  // デジタル出力ピン（LED）
  pinMode(13, OUTPUT);
```

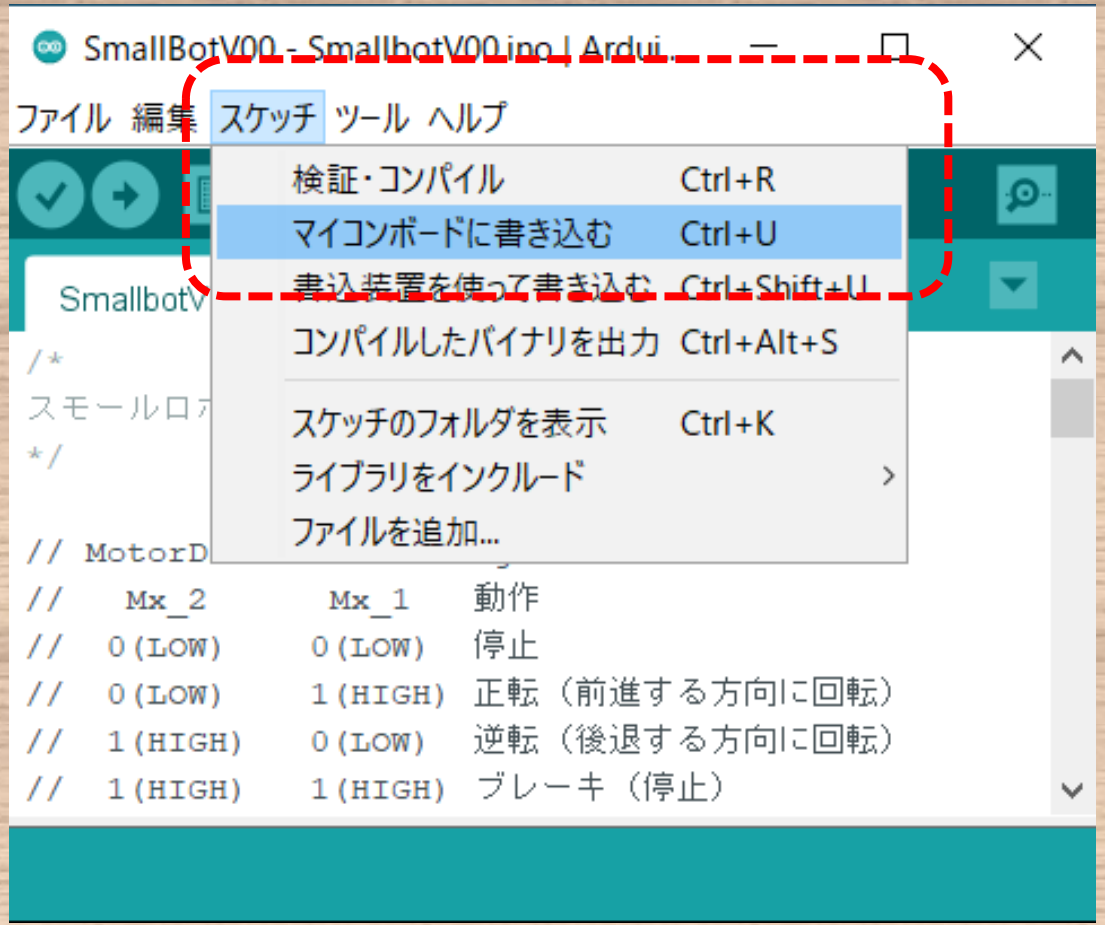
```
// 右モーターの速度（0～255）
  analogWrite(6, 80);
// 左モーターの速度（0～255）
  analogWrite(5, 120);
// 右モーター正転
  digitalWrite(4, 1);
  digitalWrite(9, 0);
// 左モーター正転
  digitalWrite(7, 1);
  digitalWrite(8, 0);
}
```

速度 80

速度 120

プログラム本体はすべて半角文字（スペースも半角！）

ビルド＆書き込み＆実行

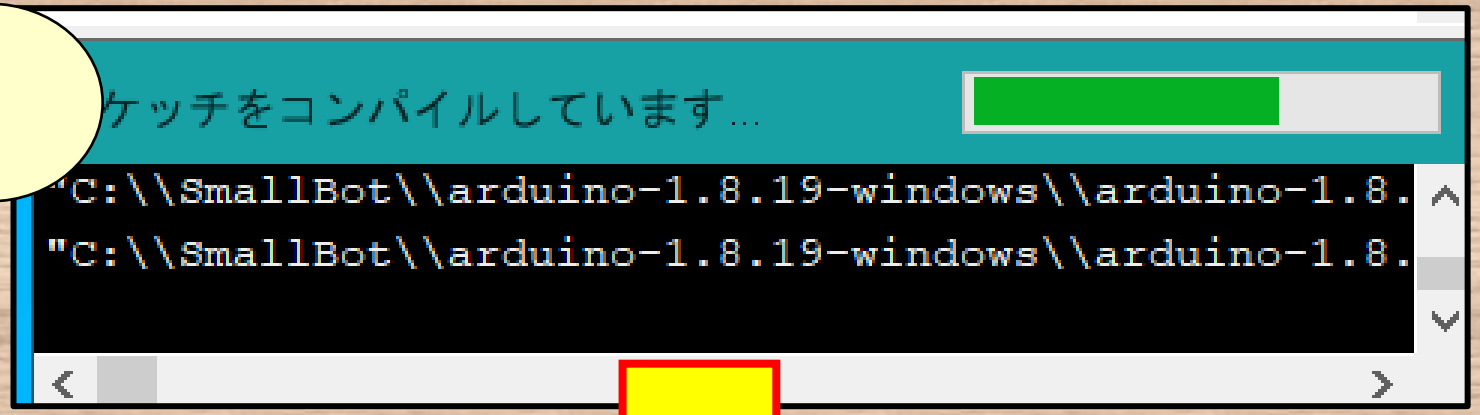


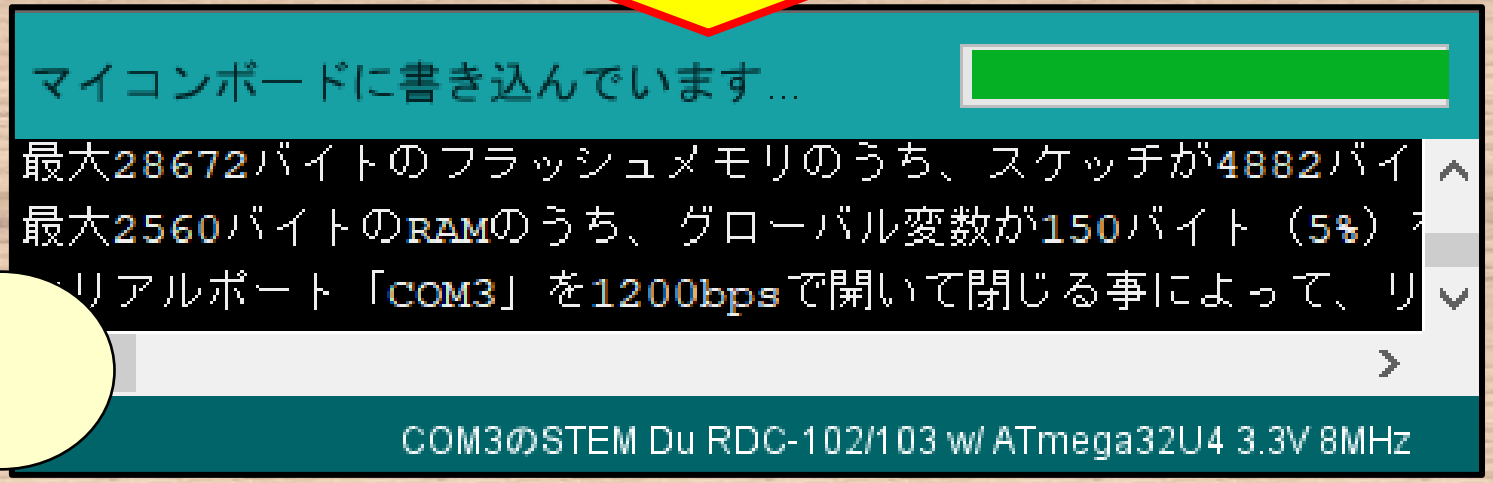
接続したSmallBotに書き込みを行う

ビルドも自動的に行われる

書き込み実行

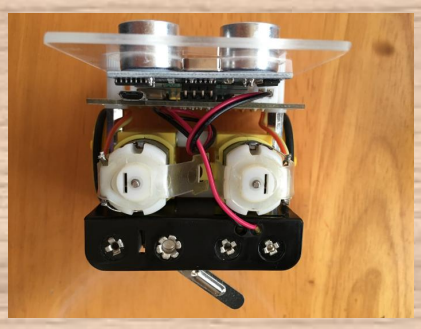
ビルド中

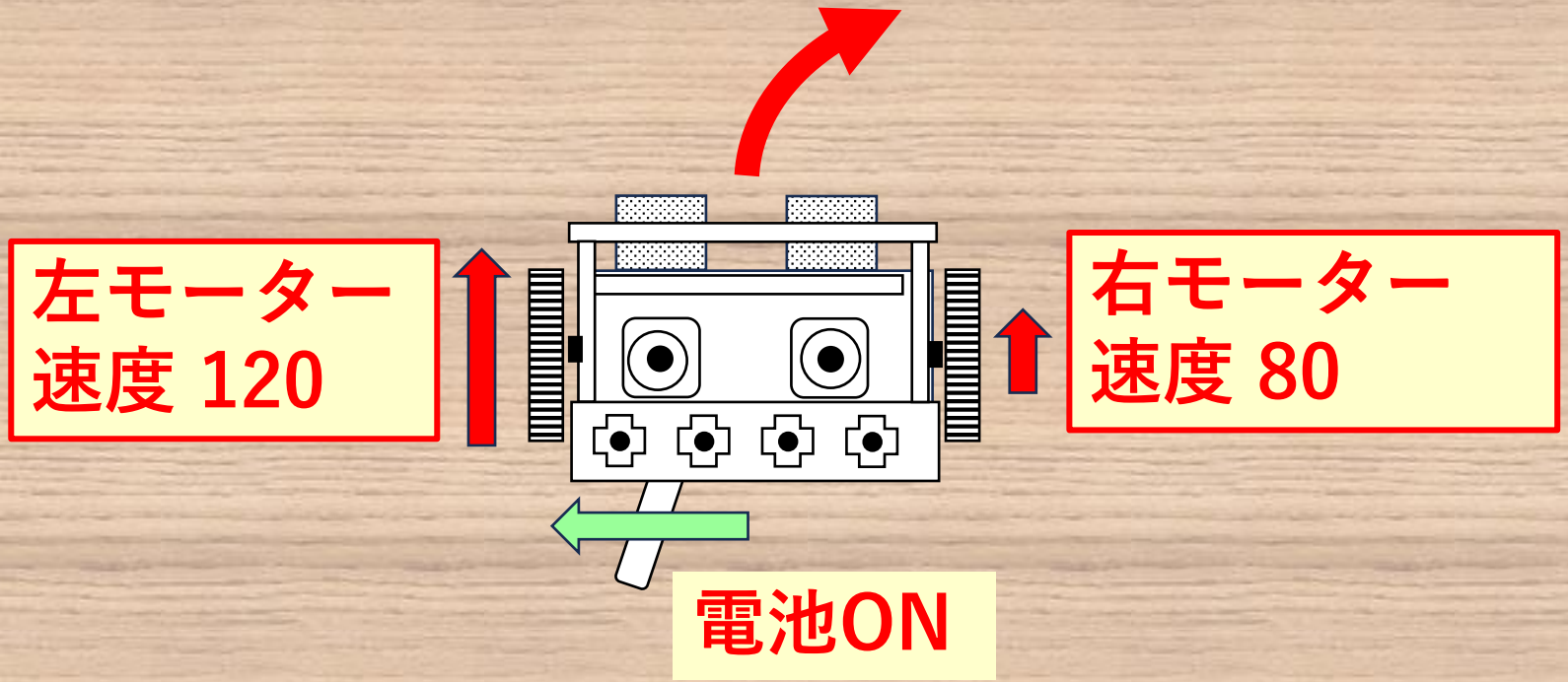




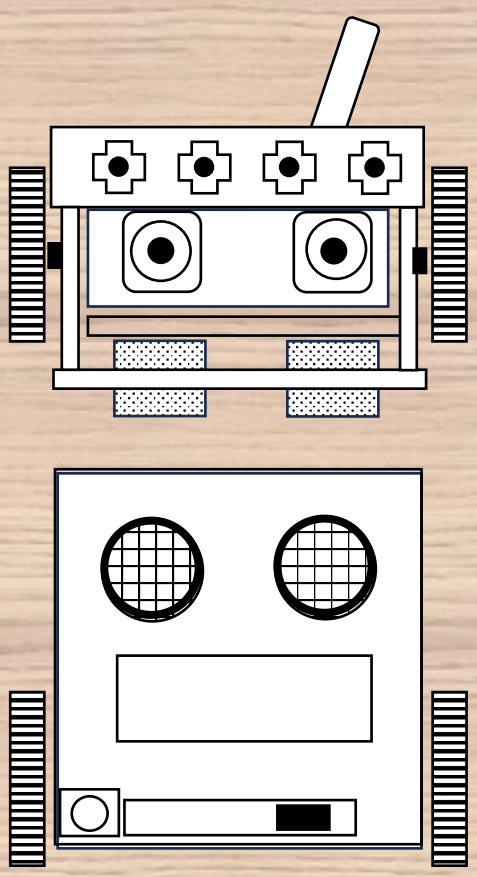
書込完了

時計回りに回転する





練習



プログラムを書き換えて
いろいろな動きをさせてみよう

モーターを逆回転等させてみよう
(00:停止 11: ブレーキ)

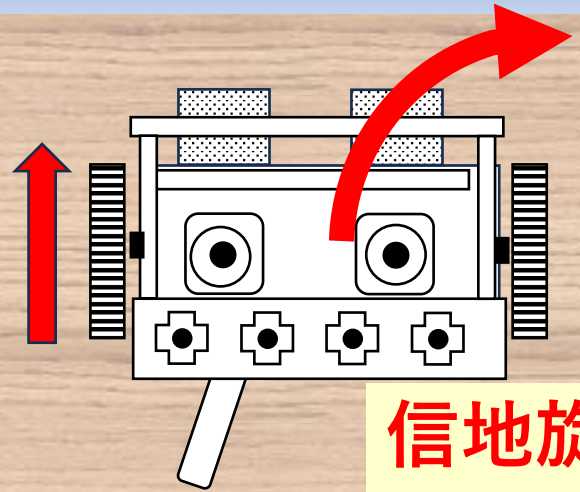
```
void setup() {
  // モータ関係の出力ピン
  pinMode(4, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(5, OUTPUT);
  // デジタル出力ピン（LED）
  pinMode(13, OUTPUT);
```

```
// 右モーターの速度（0～255）
  analogWrite(6, 80);
// 左モーターの速度（0～255）
  analogWrite(5, 120);
// 右モーター逆転
  digitalWrite(4, 0);
  digitalWrite(9, 1);
// 左モーター逆転
  digitalWrite(7, 0);
  digitalWrite(8, 1);
}
```

プログラム本体はすべて半角文字（スペースも半角！）

旋回

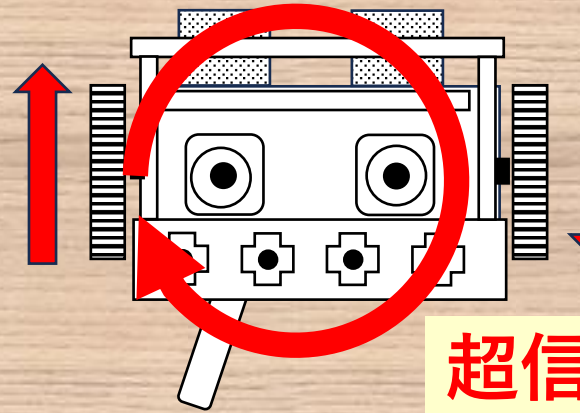
左モーター
速度 120



右モーター
速度 0

信地旋回

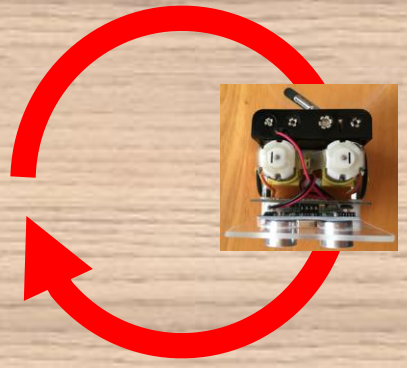
左モーター
速度 120



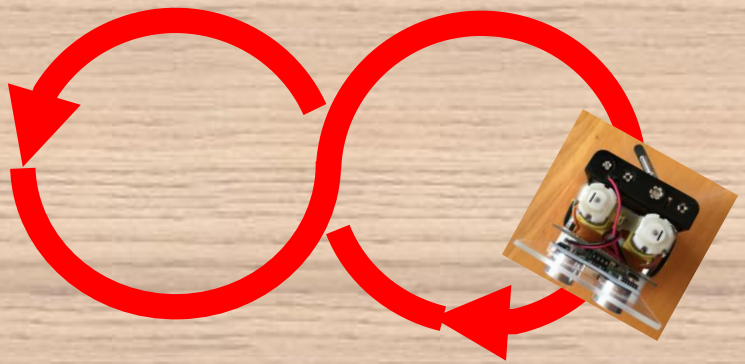
右モーター
速度 120

超信地旋回

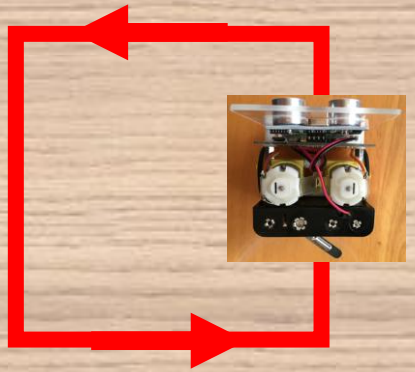
いろいろな動きをさせてみよう



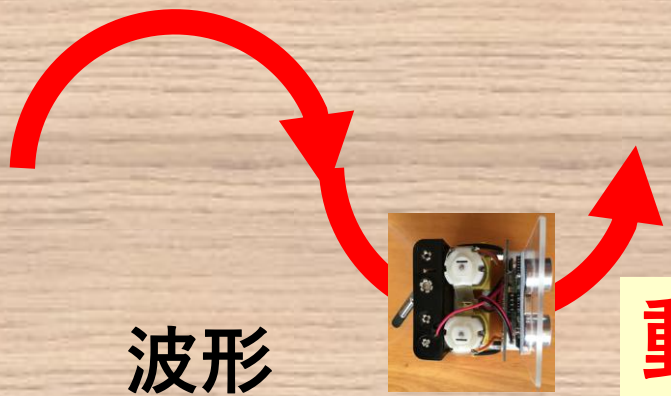
円



8の字

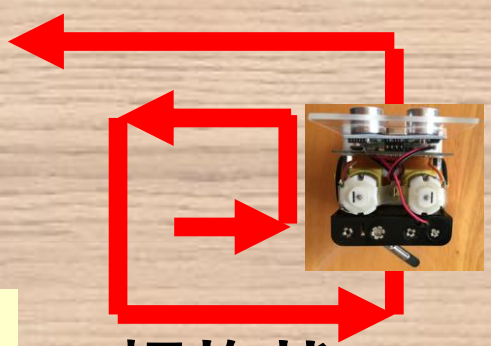


正方形



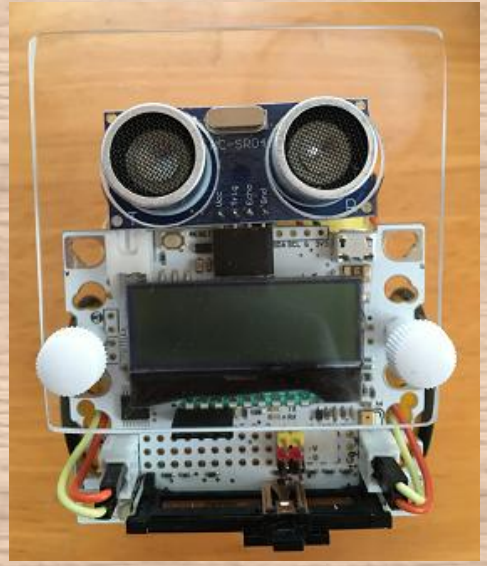
波形

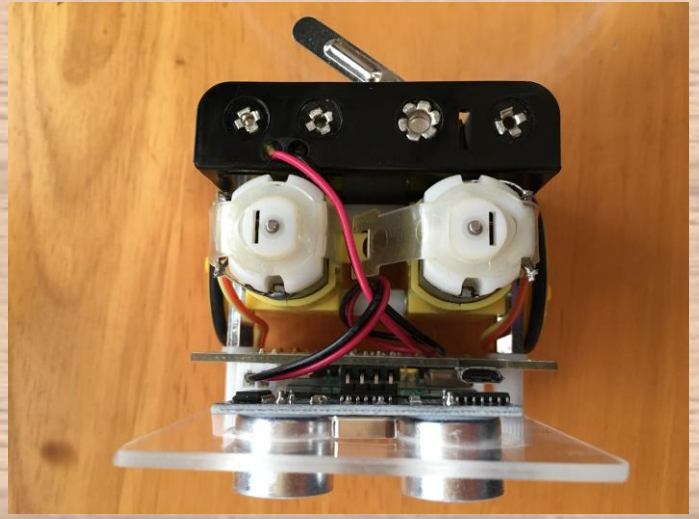
動作パターン例



螺旋状

プログラム（ライブラリ）について





ライブラリ関数の呼び出し

```
void setup() {
  // モータ関係の出力ピン
  pinMode(4, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(6, OUTPUT);

  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(5, OUTPUT);

  // デジタル出力ピン（LED）
  pinMode(13, OUTPUT);
```

```
// 右モーターの速度（0～255）
  analogWrite(6, 80);
// 左モーターの速度（0～255）
  analogWrite(5, 120);
// 右モーター正転
  digitalWrite(4, 1);
  digitalWrite(9, 0);
// 左モーター正転
  digitalWrite(7, 1);
  digitalWrite(8, 0);
}
```

ライブラリ関数などの説明

https://www.arduino.cc/reference/en/ が開く

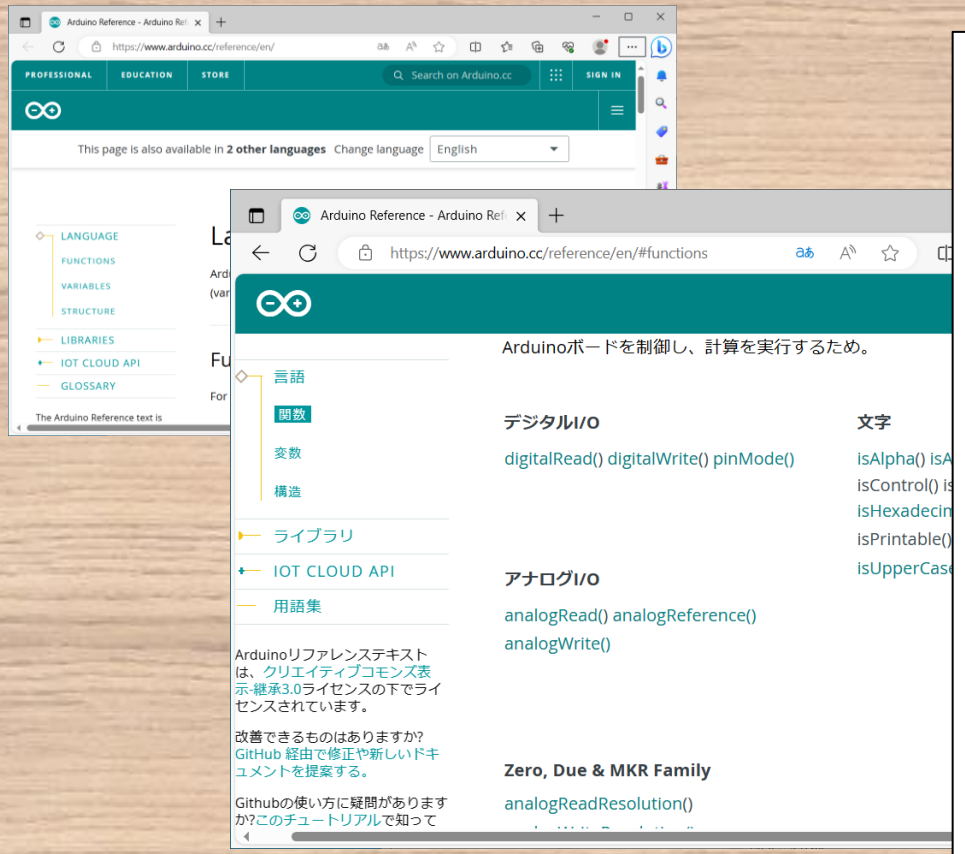
日本語の説明ページ例

http://www.musashinodenpa.com/arduino/ref/index.php
など

[arduino リファレンス]で検索してみると良い

※：公式なものではない（間違っている可能性も）

翻訳機能を活用しても良い



デジタルI/O

digitalRead() digitalWrite() pinMode()

翻訳ミスがあるかもしれないので
原文（英語）も確認しておこう

アナログI/O

analogRead() analogReference()

analogWrite()

ライブラリ：まずはここから

pinMode(pin,mode);
　ピン（端子）のモード設定

digitalWrite(pin, value);
　デジタルピンの出力設定

digitalRead(pin);
　デジタルピンの読込み

analogWrite(pin,value);
　アナログピンの出力設定

analogRead(pin);
　アナログルピンの読込み

delay(ms);
　指定した時間ウェイト
（1/1000秒：ミリ秒単位）

pulseIn(pin, value);
　パルス幅計測（μ秒単位）

delayMicroseconds(us);
指定した時間ウェイト（μ秒単位）

pinMode(pin, mode);
ピン（端子）のモード設定

pin：ピン（端子）番号
mode：動作モード
- **INPUT**
  入力
- **OUTPUT**
  出力
- **INPUT_PULLUP**
  入力(無接続時にはHIGH)

記述例：

```
pinMode(4, OUTPUT);
pinMode(11, INPUT);
pinMode(12, INPUT_PULLUP);
```

pinMode()で設定しないとき（デフォルト）はINPUTになっている

digitalWrite(pin, value);
デジタルデータ（HIGH/LOW）出力

・pin:ピン(端子)番号
・value：出力値
　・0または1

pinMode(xxx,OUTPUT);
で出力設定したピン(端子)の状態設定

0: LOW　電圧が低い（　0V）状態
1: HIGH 電圧が高い（3.3V）状態

例：

```
// 右モーター正転
digitalWrite(4, 1);
digitalWrite(9, 0);
delay(1000);
// 右モーター逆転
digitalWrite(4, 0);
digitalWrite(9, 1);
// LED点灯
digitalWrite(13, 1);
```

digitalRead(pin);
デジタルデータ（HIGH/LOW）入力

・pin: ピン（端子）番号

pinMode(xxx,INPUT);
で出力設定したピン(端子)の状態読込み

0: LOW 電圧が低い(0V) 状態
1: HIGH 電圧が高い(3.3V) 状態

例：

```
int swval;
swval=digitalRead(12);
if (swval == 0) {
  digitalWrite(13, 1);
} else {
  digitalWrite(13, 0);
}
```

analogWrite(pin, value);
PWM(パルス幅変調)出力

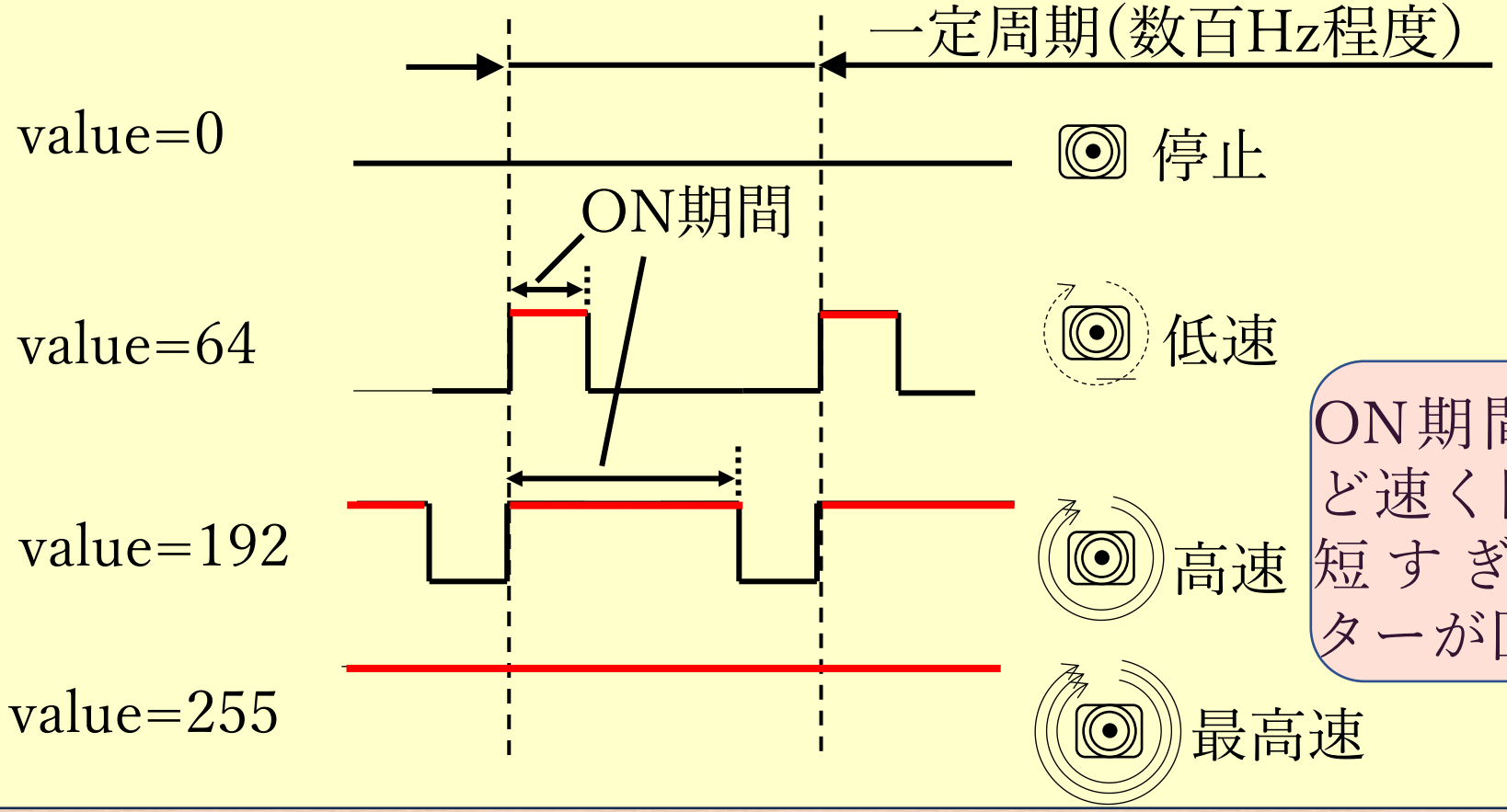
- pin: ピン（端子）番号
- value: 出力パルス幅
  設定値: 0～255

pinMode(xxx,INPUT);
で出力設定したピン(端子)の
1（HIGH)パルス幅比率を256
段階で設定

例：

```
pinMode(6, OUTPUT);
// 出力幅のHigh比率は120/255
analogWrite(6, 120);
```

analogWrite()による出力変化とモーターの回転



ON期間が長いほど速く回転するが、短すぎるとモーターが回らない

analogRead(pin);
ピン（端子）の電圧を読み込み

・pin: ピン（端子）番号

pinMode(xxx,INPUT);
で出力設定したピンの電圧
(0～3.3V)を デジタル値で返す
戻り値の範囲は0～1023
端子電圧=3.3÷1024×戻り値

例：

```
sliderValue = analogRead(A3);
if (analogRead(A2) > 200) {
 ・ ・ ・
}
```

スライダの位置や、光センサ（フォトセンサ）による
明るさの判断などに利用している

delay(ms);
1m秒単位でウェイト（動作停止）

- ms: ウェイトする時間
  (単位はm秒）

例：

```
digitalWrite(13,1); // LED ON
delay(500); // 500msウェイト
digitalWrite(13,0); // LED OFF
delay(500); // 500msウェイト
```

pulseIn(pin, value, (Timeout));
パルス幅計測(μ秒単位）

- pin: ピン（端子）番号
- value: 1 (HIGH)または
  0 (LOW)
- Timeout:タイムアウト
  μ秒単位で指定(省略可)
- **value**
  1: 入力が0から1になった後、0に戻るまでの時間
  0: 入力が1から0になった後、1に戻るまでの時間

例:

```
pinMode(11, INPUT);
duration=pulseIn(11, HIGH);
```

delayMicroseconds(us);
指定した時間ウェイト（μ秒単位）

- us: ウェイトする時間
  (μ秒単位）

例：

```
// 2μ秒ウェイト
delayMicroseconds(2);
// 1000μ秒(=1m秒）
// ウェイト
delayMicroseconds(1000);
```