

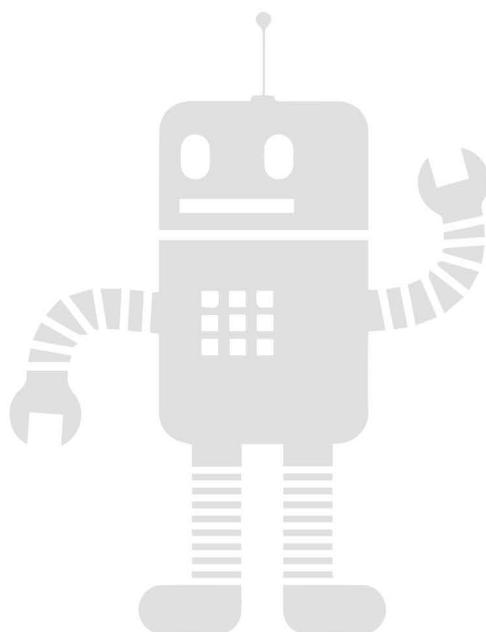


2015/03/10版

汎用計測ツール

*Measurement*  
KOBE KOSEN LAB. WORK in DENSHI. ENG.

取扱説明書



# 目次

<b>1. はじめに</b>	<b>2</b>
1.1 Measurementの目的	2
<b>2. 準備</b>	<b>3</b>
<Arduino編>	<b>3</b>
2.1 PCとArduiunoの接続	3
2.2 Arduino IDEの起動	3
2.3 Arduinoへの書き込み手順	4
<Processing編>	<b>6</b>
2.4 Processing IDEの起動	6
2.5 スケッチの実行	7
<b>3. ツールの使用方法</b>	<b>8</b>
3.1 タイトル画面	8
<Measuring Mode>	<b>9</b>
3.2 各部説明	9
3.3 抵抗測定時 (A3) のみの機能について	12
3.4 チェックモードとは?	14
<Browse Mode>	<b>15</b>
3.5 Browse Modeとは?	15
<b>4. 実践</b>	<b>16</b>
4.1 モータからの出力電圧の計測	16
4.2 曲げセンサの抵抗値 (曲がり具合) の計測	16
<b>5. さいごに</b>	<b>18</b>

# 1. はじめに

## 1.1 Measurementの目的

Measurementは、工学を専門に学ぶ人のためのプログラミング学習、および風車からの発電電圧を測定する等の実践的実験実習のための汎用計測ツールです。初期状態では電圧値、抵抗値を測定・表示することができます。

本ツールは、パソコン、Arduino、Processingの環境があれば動作します。そのため、市販されている電圧計よりも安価であるのが特徴です。また、自らプログラムを変更することで様々なセンサ値を測定することが可能です。そのため、自由度が高いのも特徴です。

本説明書では、このツールの使用方法を説明します。（なお、Arduino、Processingの環境は構築済みとします。）

### ツールを使用する上での注意点



以下に、ArduinoやパソコンのUSBポートが損傷してしまう危険な接続方法等を説明します。絶対に行わないでください。

1. 入出力ピンに5.5V以上の電圧を供給する。
2. RESETピンに13V以上の電圧を供給する。
3. 入出力ピン同士をショートさせる（お互いに接続させる）。※1
4. VINピン、もしくはGNDピンに電源を逆流させて接続する。
5. **VINピンとGNDピンをショートさせる。**※2
6. 5Vピンに6V以上、3V3ピンに3.6V以上の電圧を供給する。※3
7. 給電されている状態で、5Vピン、3V3ピンに電圧を供給する。

※1：片方のピンが出力モードでHIGH、もう片方がLOWに設定されている場合。

※2：VINピンに電圧入力がある場合。

※3：外部電源による給電やUSBによる給電がなされていない場合。

**赤文字**は、非常に危険な接続方法を表しています。これらはUSBポートやボード上のマイコン、多くの素子を破壊したり、ボードが溶けてしまう恐れがあります。

また、ピンの名前はボード上に記載されていますのでそれを参考にしてください。（Arduinoによって異なる場合があります）

なお、入出力ピンとは、ボード上の ANALOG IN (A0~A5) と DIGITAL (0~13) を指します。（Arduinoによって異なる場合があります）

# 2. 準備 | Arduino編

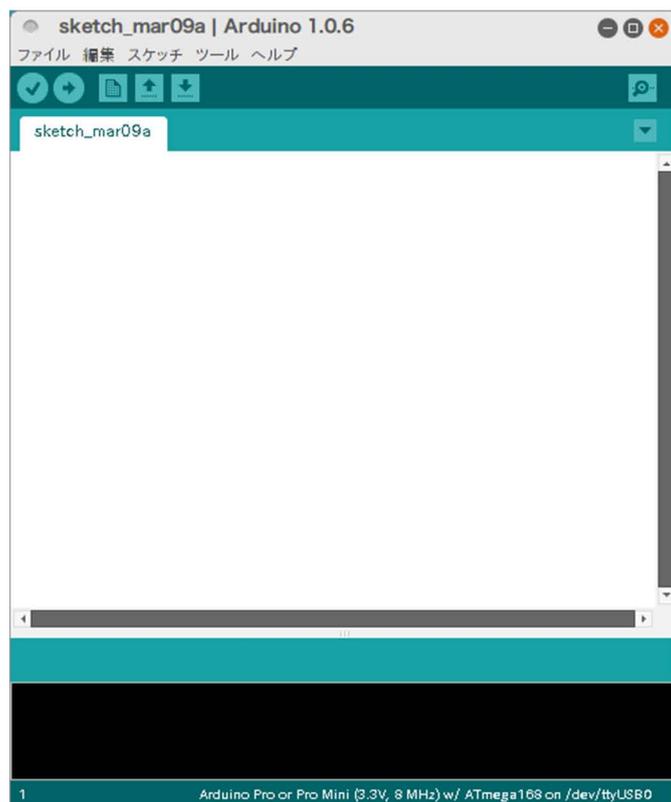
## 2.1 PCとArduinoの接続

パソコンとArduinoの接続には、USBケーブルを使用します。USBケーブルは、片方がAタイプのオス、もう片方がBタイプのオスのものを使用します。（Arduinoによって異なる場合があります）

USBケーブルで接続すると、Arduinoは自動的にUSB給電モードになり、5Vピン、3V3ピンにも電圧が供給されます。そのため、VINに電圧を入力したり、外部電源による給電を行う必要は特にありません。**特に、この状態で5Vピンや3V3ピンに電圧を供給すると、素子が破壊されてしまうおそれがあるため、絶対に行わないでください。**

なお、ドライバが必要な場合は、指示に従いインストールします。

## 2.2 Arduino IDEの起動



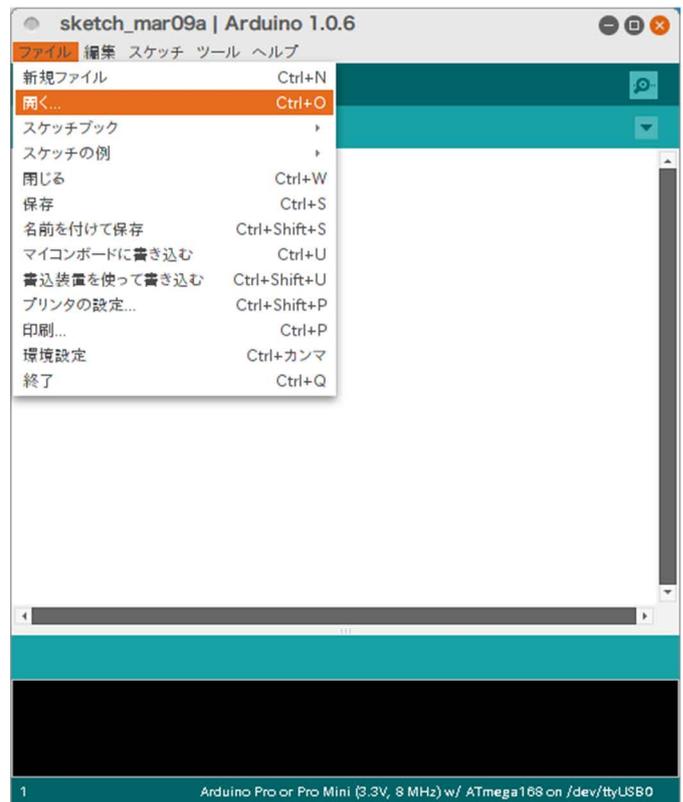
Arduinoを起動すると、このような画面がでてきます（画像はLinux Mintのものです）。この白い枠部にプログラムを書いていきます。Arduinoのプログラムのことを「**スケッチ**」と呼んだりします。

今回は、計測ツールのためのArduino用のスケッチをあらかじめ用意してあるので、これを用いてArduinoのマイコンに書き込む方法を説明します。

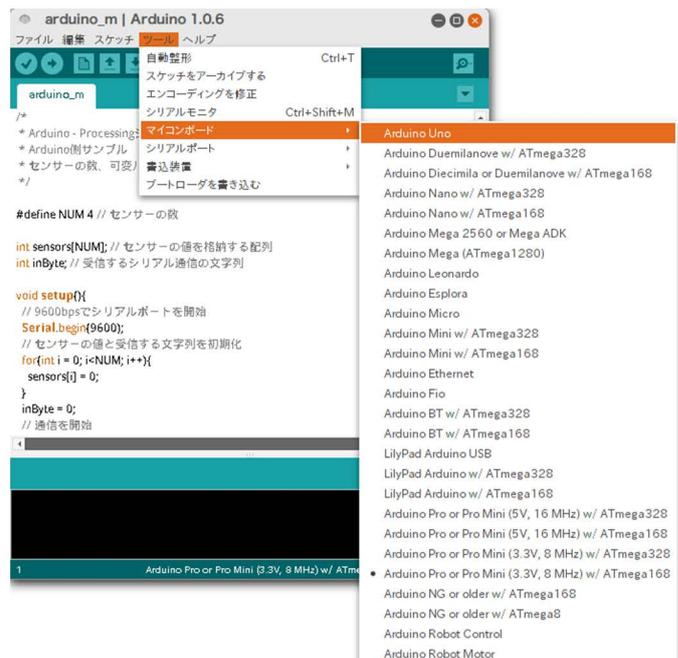
# 2. 準備 | Arduino編

## 2.3 Arduinoへの書き込み手順

- ①まず、ファイル>開く...を選択し、測定ツール用のスケッチファイル「arduino\_m.ino」を開きます。ファイルを開くと、画面白枠にそのプログラムが表示されます。



- ②次に、マイコンボードを選択します。使用するArduinoの種類を確認し、ツール>マイコンボードより選択します。本説明では「Arduino Uno」を使用しています。



※ 「マイコンボード」などの項目名は、使用するIDEのバージョンによって違う可能性があります。

## 2. 準備 | Arduino編

- ③次に、ツール>シリアルポートより、シリアルポートを選択します。Arduinoに割り当てられたポート番号（図は/dev/ttyUSB0となっていますが、環境によって変動します。）を選択します。もし、シリアルポートが選択できない場合、USB接続が認識されているか、デバイスドライバが正常にインストールされているか等を確認してください。



- ④最後に、スケッチをArduinoにアップロードします。円の中に右向きの矢印が書かれているボタン（) をクリックすると、スケッチのコンパイルが始まり、コンパイル終了後、自動でマイコンボードに書き込まれます。書き込みに失敗する場合、シリアルポートの設定を確認してください。

これで、Arduinoの準備が完了しました。



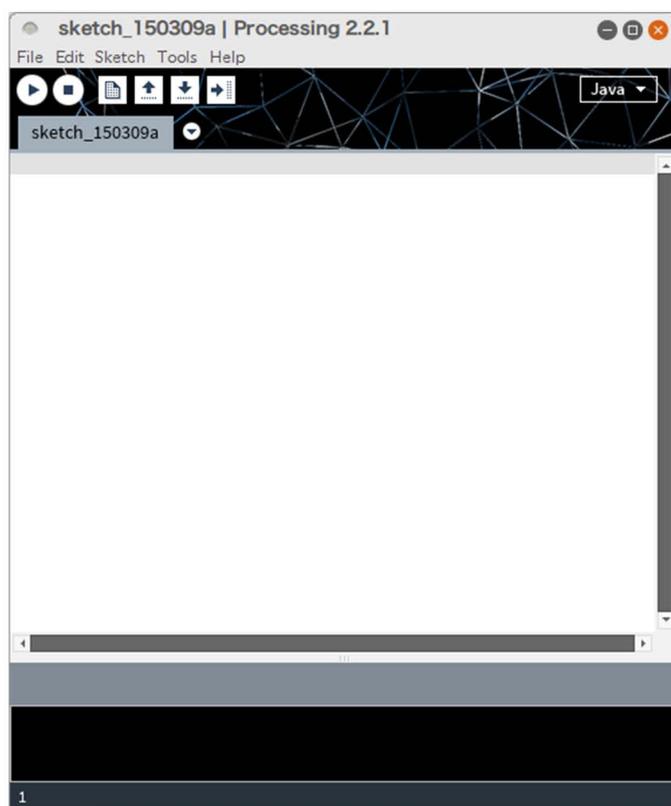
## 2. 準備 | Processing編

### 2.4 Processing IDEの起動

ツールのメイン部分となるのがProcessingです。このツールは、ArduinoとProcessingでリアルタイムのデータ送受信を行っています。今回用いている通信方法は「ハンドシェイク通信」と呼ばれますが、詳しくは割愛します。

ProcessingIDEを起動すると、下の図のような画面になると思います。Arduino同様、画面の白枠にプログラムを書いていきます。

今回は、計測ツールのためのProcessing用のスケッチを用います。

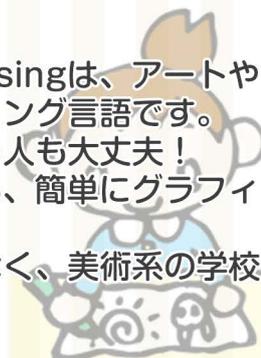


#### Processingって何!?

堅苦しいイメージのある「プログラミング」ですが、Processingは、アートやビジュアルデザイン、動きや美しさを重視した大胆なプログラミング言語です。

そうは言ってもプログラミングは苦手なんだよな・・・という人も大丈夫！ Processingの特徴は、プログラミングの知識がさほどなくても、簡単にグラフィック表現が可能であることが挙げられます。

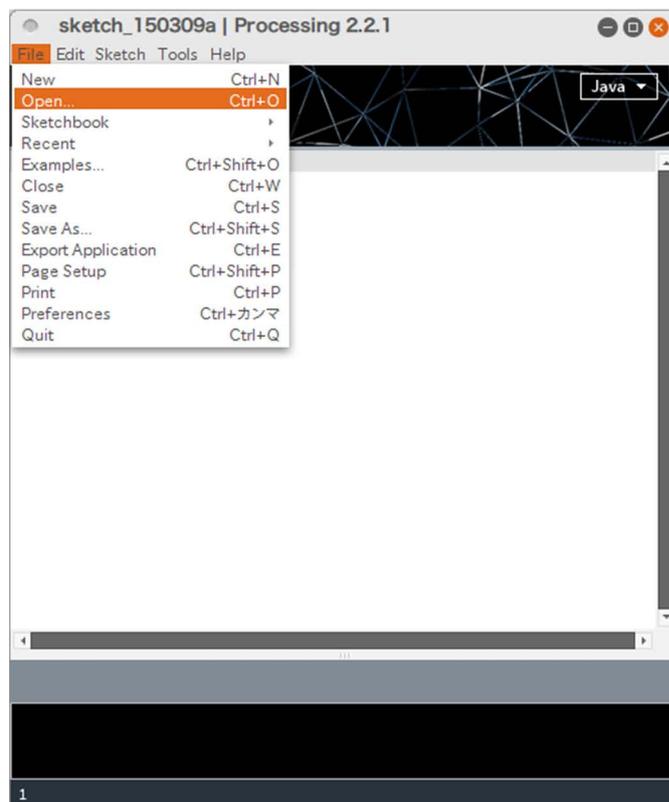
その簡単さ、表現力の高さから、工業系の大学や高専だけでなく、美術系の学校でのデザインツールとしても用いられてたりします。



# 2. 準備 | Processing編

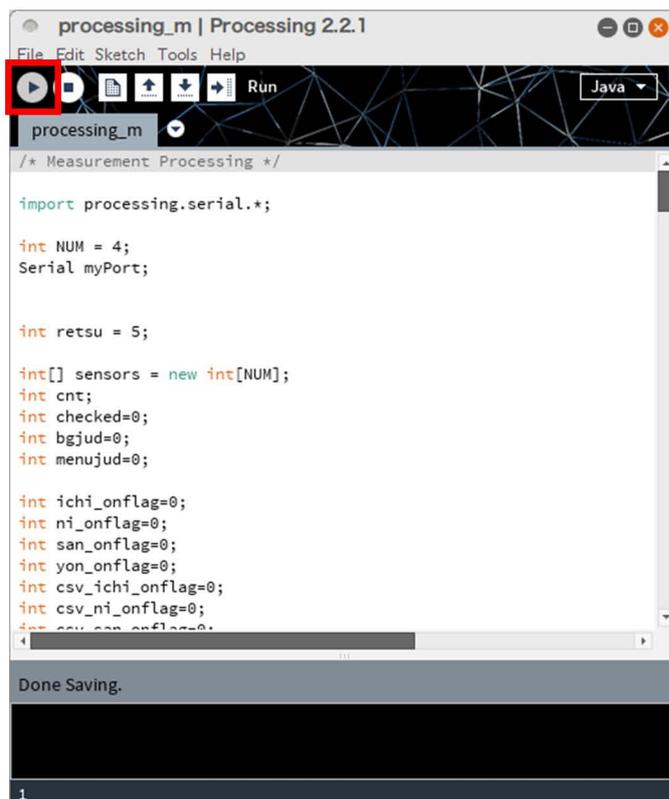
## 2.5 スケッチの実行

- ①まず、File>Open...を選択し、測定ツール用のスケッチファイル「processing\_m.pde」を開きます。ファイルを開くと、画面白枠にそのプログラムが表示されます。



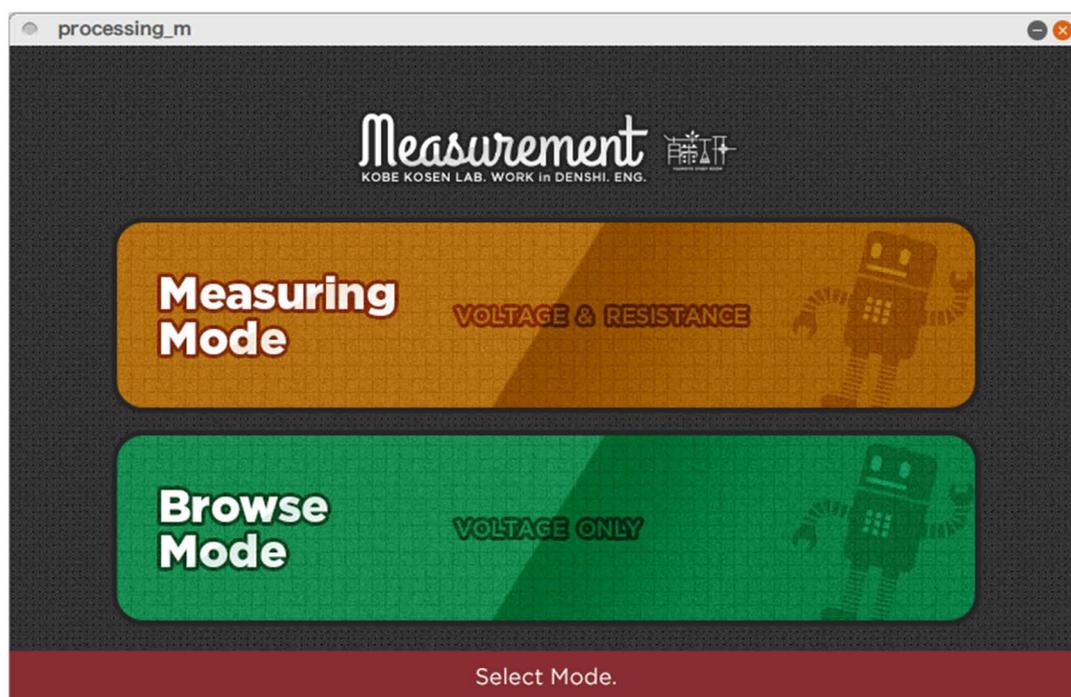
- ②あとは実行するだけです。円の中に再生マークがかかれたボタン (▶) をクリックすると、プログラムが実行され、計測ツールが起動します。エラーが発生する場合、Arduinoが接続されているかを確認してください。

これで、準備が整いました。次ページより、このツールの使用方法について説明します。



# 3. ツールの使用方法

## 3.1 タイトル画面

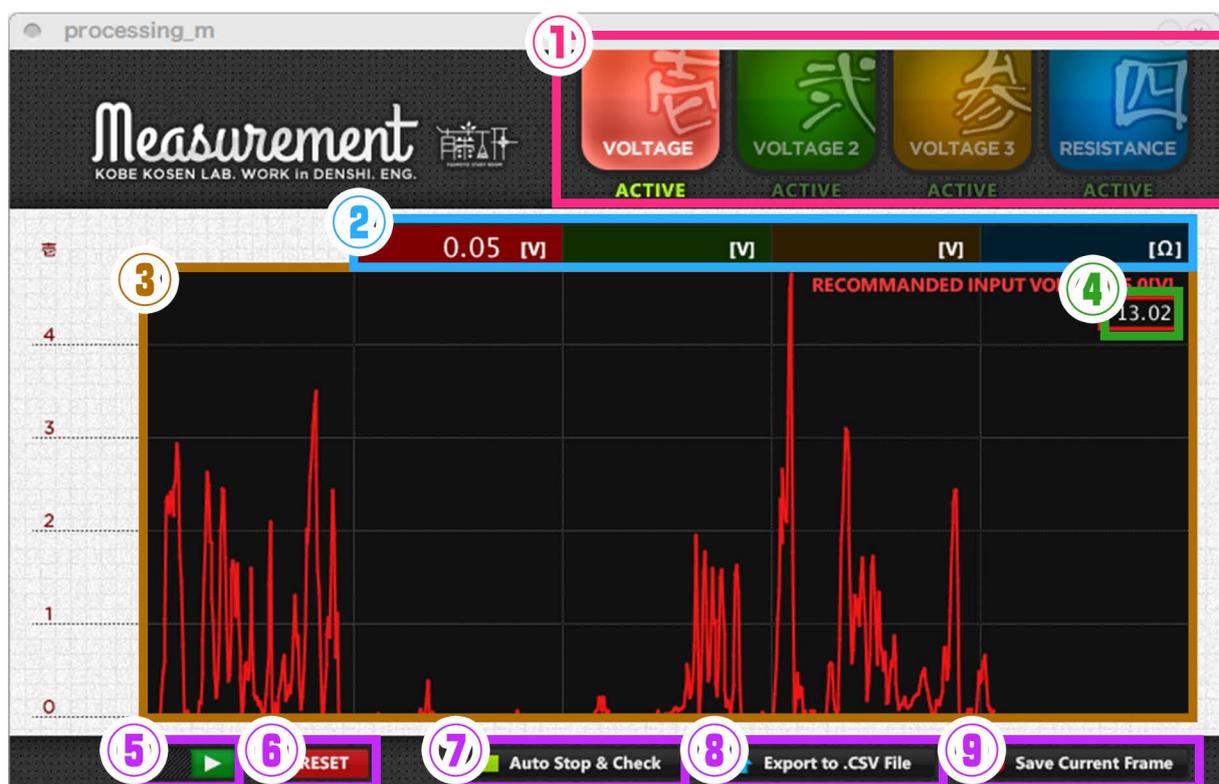


プログラムを実行すると、上の図のような画面が表示されると思います。これがこのツールのタイトル画面です。このツールには2つのモードがあり、1つは電圧・抵抗値などを計測する「Measuring Mode」。もう1つは、Measuring Modeで保存したcsvを読み込んで表示させることができる「Browse Mode」です。

次のページより、この2つのモードについて詳しく説明します。

# 3. ツールの使用方法 | Measuring Mode

## 3.2 各部説明



### ①測定ON / OFF切り替えボタン

計測対象を接続しているポートに対応するボタンをONにします。同時に4つまで計測し、グラフ表示が可能です。なお、対応表は以下の通りです。

ボタン	対応しているArduinoの入出力ポート	用途
	ANALOG IN 0番ポート (A0)	電圧測定
	ANALOG IN 1番ポート (A1)	電圧測定
	ANALOG IN 2番ポート (A2)	電圧測定
	ANALOG IN 3番ポート (A3)	抵抗測定

# 3. ツールの使用方法 | Measuring Mode

## ②計測値表示部

ポートに入力されている計測値がリアルタイムで表示されます。それぞれ各測定ON/OFF切り替えボタンの色に対応しています。

## ③グラフ描画部

ポートに入力されている電圧がリアルタイムでグラフ化される、オシロスコープのようなものです。線の色はそれぞれ各測定ON/OFF切り替えボタンの色に対応しています。なお、グラフが画面の右端まで行くとグラフが消去され、繰り返し左端から描画し始めます。

## ④経過時間表示部

計測時間を表示しています（秒）。一定の周波数を出力する計測対象に対して周期を求めたいときなどに便利です。なお、グラフが画面の右端まで行くと、経過時間は0にリセットされ、繰り返しカウントし始めます。

## ⑤計測の再開／一時停止

計測の再開および一時停止を行うボタンです。一時停止中は**チェックモード**（P.06参照）になります。

## ⑥リセットボタン

計測のリセット（画面のリフレッシュ）が行われます。グラフや経過時間はリセットされ、グラフは左端から描画、経過時間は0秒からカウントし始めます。

## ⑦オートモードボタン

このボタンをクリックすると、グラフが右端まで行くと自動で一時停止し、チェックモードになります。そのほか、データのcsv出力（次ページ参照）、画面キャプチャ（次ページ参照）が自動で行われます。

一時停止状態から抜けたいときはリセットボタンを押してください。

# 3. ツールの使用方法 | Measuring Mode

## ⑧ csv出力ボタン

計測データをcsv形式で出力します。このボタンをクリックした時期にかかわらず、グラフが右端まで到達した段階で画面左端から右端までのデータが出力されます。**(画面途中から、および途中までのデータ出力は原則できません)**なお、データ出力後、画面のリフレッシュが行われます。

連続でデータをとりたいときは、リフレッシュが行われてから再度このボタンをクリックすることで、同様に出力されます。

なお、データはこのツールのpdeファイルがある場所に生成され、そのファイル名は「data\_Y-M-D-h-m-s.csv」となります。



## ⑨ 画面キャプチャボタン

ツール全体のキャプチャを行います。画像形式はpngです。csv出力と同じく、このツールのpdeファイルがある場所に生成され、そのファイル名は「data\_Y-M-D-h-m-s.png」となります。

なお、タイトル画面に戻りたいときは、Measurementロゴをクリックします。ただし、測定ON/OFF切り替えボタンをすべて「OFF状態 (Activeが消灯し、ボタンが暗い状態)」にしなければいけません。

※Y:年、M:月、D:日、h:時、m:分、s:秒。たとえば、2015年5月17日14時0分33秒に保存された場合、data\_2015-5-17-14-0-33.(拡張子)となります。

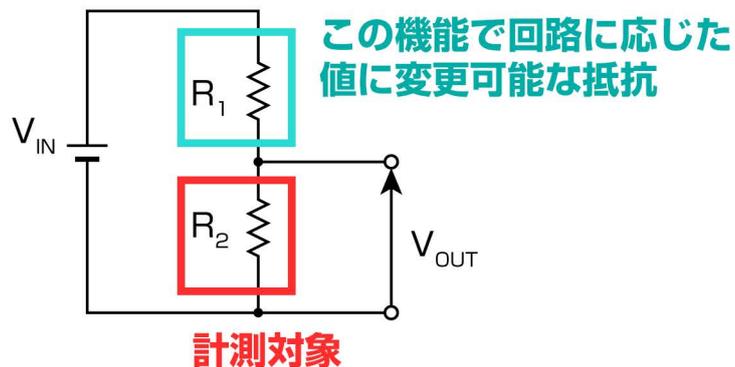
# 3. ツールの使用方法 | Measuring Mode

## 3.3 抵抗測定時（A3）のみの機能について



### 分圧抵抗変更機能

このツールで抵抗を測定するには、分圧回路を作成する必要があります。分圧回路とは、入力電圧に比例した出力電圧を出力する回路です。



上の図は、簡単な分圧回路です。たとえば、 $V_{IN}=5[V]$ 、 $R_1=4.7[k\Omega]$ 、 $R_2=10[k\Omega]$ とすると、 $R_2$ の両端電圧 $V_{OUT}$ は、 $R_1$ と $R_2$ の比をとればいいので、 $10/(4.7+10) \times 5 \div 3.40[V]$ となります。このツールでは、この $V_{OUT}$ と $R_1$ をもとに（ $V_{IN}$ はArduinoの5Vピンから供給するものとする） $R_2$ を割り

# 3. ツールの使用方法 | Measuring Mode

出します。その $R_1$ を柔軟に変更できるのがこの機能です。初期値は4.7[k $\Omega$ ]になっています。

たとえば回路上で $R_1$ を10[k $\Omega$ ]にしたとき、そのままだとツール内部では $R_1$ は4.7[k $\Omega$ ]として計算しているため、ツールに表示される計測結果（抵抗値）は間違ったものとなります。 $R_1$ を変更するには、単位「 $\Omega$ 」でキーボード入力を行います。10[k $\Omega$ ]に変更するには、「1」「0」「0」「0」「0」「ENTER」の順でキーボードを押すと、小窓内の4700となっていた部分が10000に変わり、正しい $R_1$ の値が適用されます。このことによって、ツールに表示される計測結果（抵抗値）も正しいものとなります。



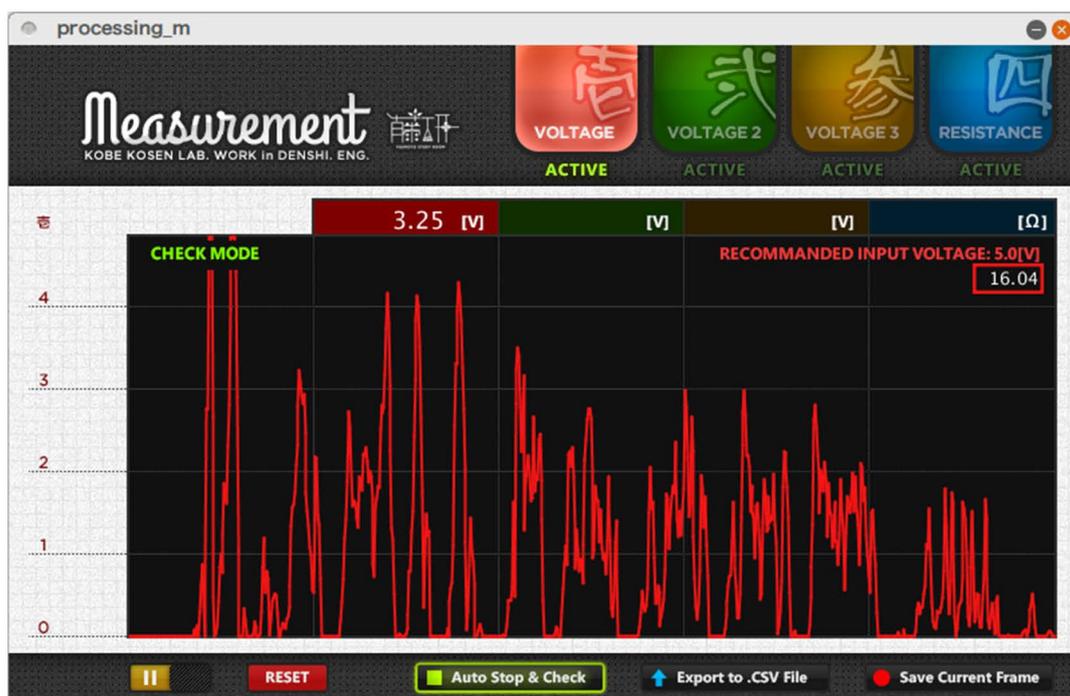
抵抗値測定機能では、計測値表示部のサイズの関係で、1000[ $\Omega$ ]以上になると「k  $\rightarrow$   $10^3$ 」、1000[k $\Omega$ ]以上になると「M  $\rightarrow$   $10^6$ 」と省略表示されます。たとえば、測定結果が3400[ $\Omega$ ]のとき、3400と表示されるのではなく、3.4kと表示されます。

なお、No Inputと表示された場合、5Vが直接入力された状態（つまり、分圧できていない状態）を示します。回路の確認を行って下さい。

# 3. ツールの使用方法 | Measuring Mode

## 3.4 チェックモードとは？

チェックモードは、グラフ描画部内のクリックした部分（時間）の電圧を、計測値表示部に表示させる機能です。緑色のCHECK MODEの文字が表示されている状態です。手動で一時停止した場合、およびオートモードボタンをクリックして自動で一時停止した後、自動的にこのモードになります。再開、もしくはRESETボタンをクリックすることで計測モードに戻ります。



さて、ここまでがMeasuring Modeの説明でした。電圧を測定する際はグラフを注視し、**グラフが描画部の天井（MAX値）に振り切った場合は出力を落とすなどの処置をしてください。**

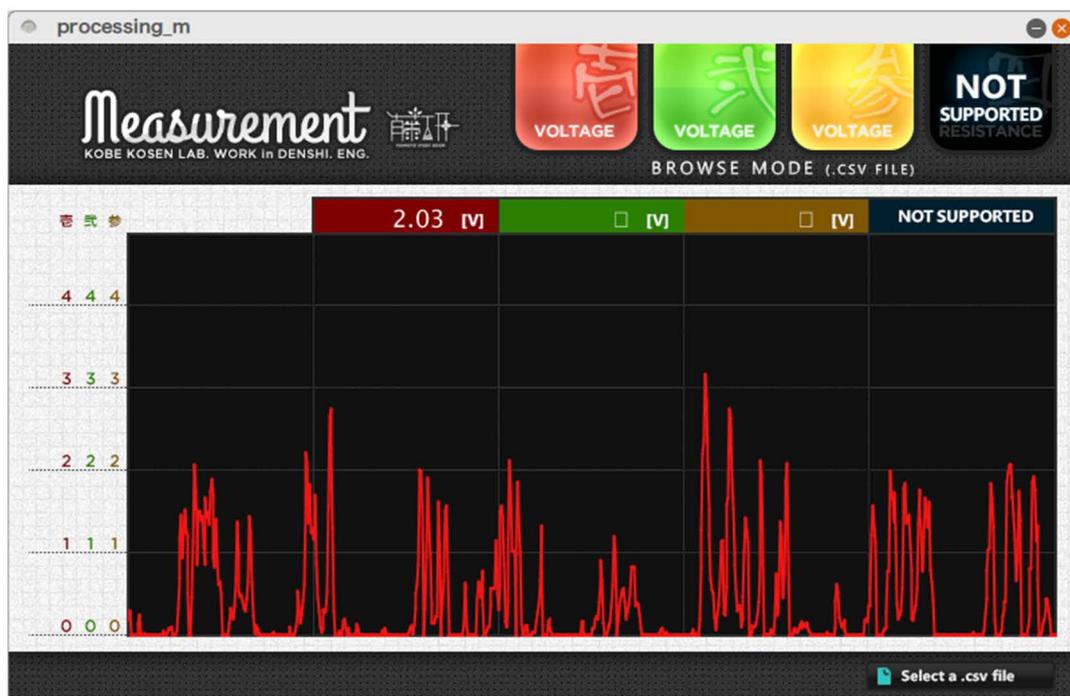
このような状態にあるにもかかわらず無視してそのまま計測を続けると、**Arduinoの素子が破壊される恐れがあります。**

続いてBrowse Modeの説明に移ります。

# 3. ツールの使用方法 | Browse Mode

## 3.5 Browse Modeとは？

Browse Modeは、Measuring Modeでcsvデータ出力したものを読み込んで表示するツールです。また、チェックモードの機能が備わっています。



まず、画面左下の「select a .csv file」をクリックします。すると、ファイル選択画面に移動しますので、測定したcsv出力データを選択します。

データを選択すると上の図のようにグラフが表示されます。また、描画部内のクリックした部分の値が計測値表示部に表示されます。

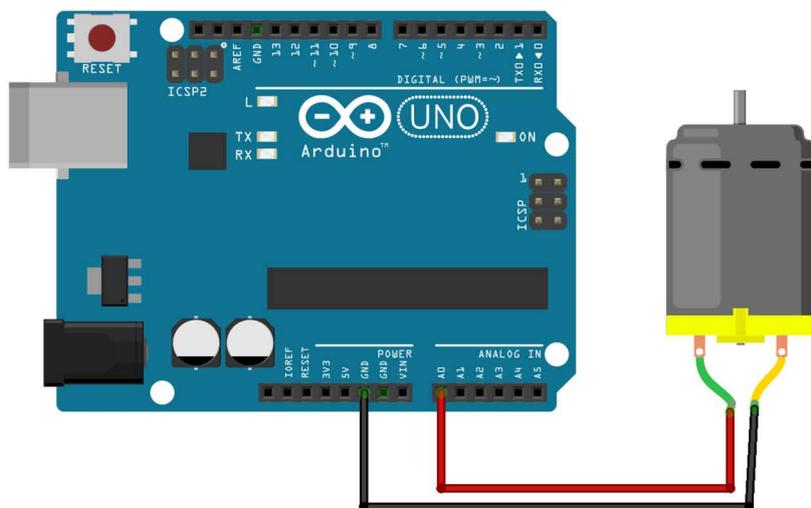
なお、Browse ModeはA3ポートの計測（抵抗値）のグラフ表示および抵抗値表示には対応していません。測定していたとしても無視されます。

Browse Modeからタイトル画面に戻りたい場合は、同様に、Measurementロゴをクリックします。

# 4. 実践

## 4.1 モータからの出力電圧の計測

モータの端子の片方をArduinoのA0ポート、もう片方をGNDに接続し、Measuring Modeで測定ON/OFF切り替えボタン「壺」をONにします。そのあと、モータを手で回転させると、電圧の変化グラフが描かれます。



もしグラフが変化しない場合、逆方向に回していないか、ポートに導線がしっかり接続されているか等を確認してください。

**注意：**速く回しすぎると、許容電圧を超えてしまい、素子の寿命を縮めてしまう恐れがあります。ゆっくり、グラフを確認しながらまわしてください。

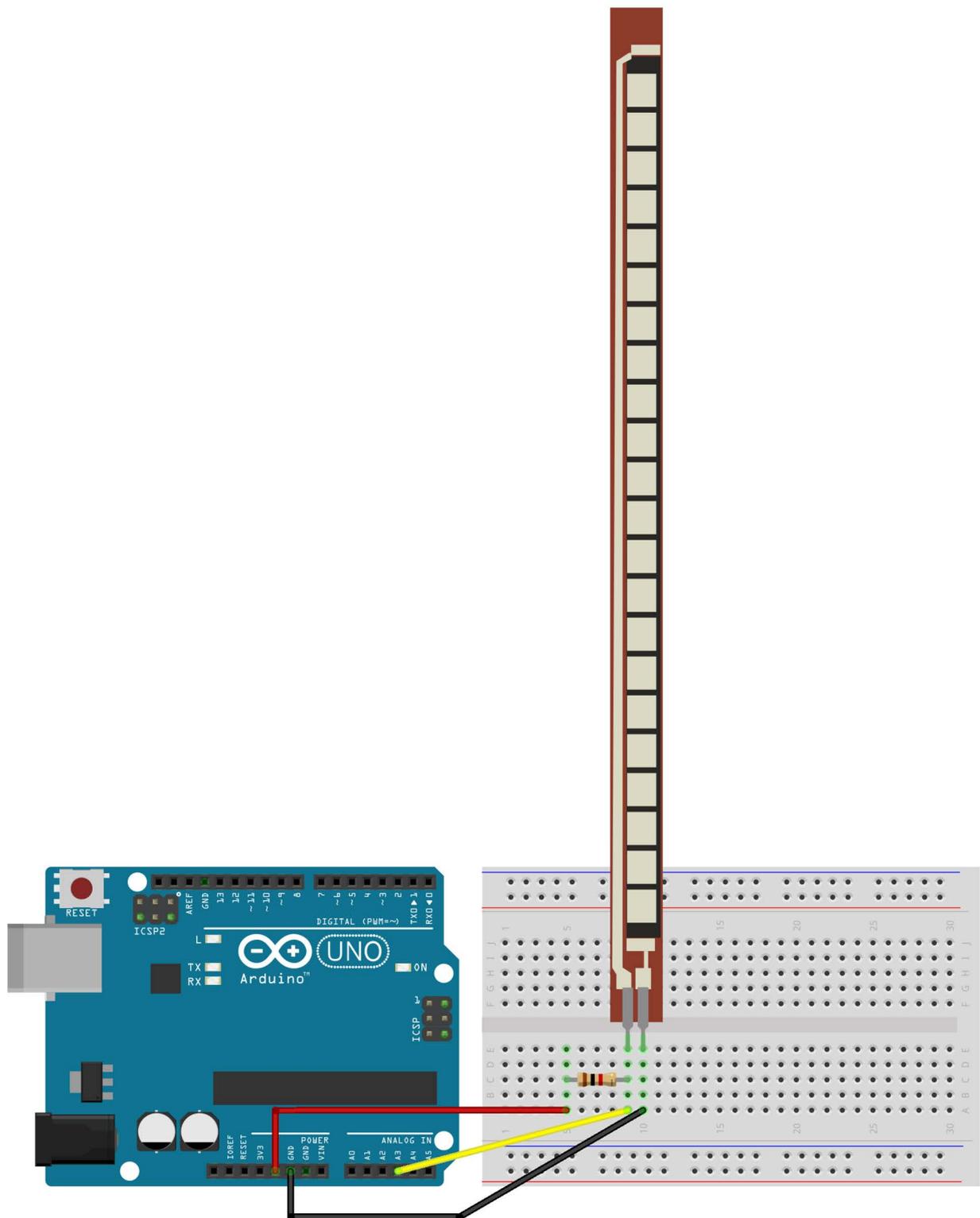
## 4.2 曲げセンサの抵抗値（曲がり具合）の計測

曲げセンサは、曲げると抵抗値が大きくなるセンサです。抵抗値を測定するには、まず分圧回路をブレッドボード上で構成します。そして、曲げセンサにかかる電圧をA3ポートに入力し、あとは5Vピン、GNDピンに次ページの図のように接続します。

図中の抵抗は4.7[k $\Omega$ ]ですが、他の抵抗を使用しても構いません（小さすぎたり、大きすぎたりしないようにしてください。）

曲げセンサを曲げると、グラフが変化し、計測値表示部に表示されている抵抗値も変化します。

# 4. 実践



**注意：**ブレッドボード上で回路を構成し終えてから、Arduinoのポートに順次接続させ、**回路の確認をしてから、Arduinoに給電を行ってください。**回路に間違いがあるまま給電を行った場合、最悪の場合素子を破壊してしまう恐れがあります。

# 5. さいごに

ツールを使用してみて、いかがでしたでしょうか。初期状態では電圧と抵抗しか測定できませんが、プログラムを自分で編集することで、さまざまなセンサ等に対応が可能です。プログラムソースにはコメントが記されていますので、ぜひそれを参考にして機能を追加してみてください。

それでは、楽しい工学ライフを！

## ※お願い

このツールは、定電圧源をもとにして正確な電圧が測定できるように調整しましたが、使用するArduinoによってずれる可能性があります。

もしずれが生じている場合、お手数ですが、「processing\_m.pde」のdraw\_graph関数内の以下の式の「62」を小数点単位で変更し、微調整を行ってください。4箇所あります。

```
      ⋮  
float analog1 = abs(ty1 - 444) / 62;  
      ⋮  
float analog2 = abs(ty3 - 444) / 62;  
      ⋮  
float analog3 = abs(ty3 - 444) / 62;  
      ⋮  
float analog4 = abs(ty4 - 444) / 62;  
      ⋮
```