

■Gotoronic

arduinoというMCU+開発環境を用い、市販で入手できる電子部品ほか材料を用いて個人で手軽に電動変速器を自作するためのソリューションを提供する。

構成：MCUボード、変速機を駆動するサーボ、サーボ固定するステー、駆動用のロッド、変速トリガを入れるスイッチ、バッテリー、ハーネス、etc...

■arduinoとは

[\[入門\]Arduinoのはじめかた](#)

いわく：一言でいうと「初心者でも簡単に扱えるマイコンボード」です。
また、オープンソース・ハードウェアと呼ばれるもので回路からソフトウェアの全てが公開されています。
このため、これまで電子工作にあまりなじみのなかった人やマイコンのプログラミングに詳しくない人でも、すぐに始める事が出来るように考えられています、このため電子工作の入門にお勧めの一つとなっています。

用語：IDE＝開発環境。各自PCにインストールする。
スケッチ＝arduino本体(MCUボード)に書き込むプログラム。

参考サイト

[Arduino 日本語リファレンス](#)

オリジナルの英文を訳した文章が主。わりとわかりやすい。

[workshop/ServoBasis/main](#)

サーボライブラリ詳説

■サーボの制御について

Gotoronicでは“writeMicroseconds(uS)”コマンドを主に用いている。
サーボに対しマイクロ秒単位で角度を指定します。標準的なサーボでは、1000で反時計回りにいっぱいまで振れます。2000で時計回りいっぱいです。
製品によっては、この範囲に収まらず700～2300といった値を取るものもあります。
※1uSecごとの移動θ目安 $180\text{deg}/(2000\text{uSec}-1000\text{uSec})=0.18\text{deg}$

Gotoronicでは1stギヤ位置をpos0として定義する。
この時サーボとしては動作最大角度に近い側になっている。
(※使うサーボ製品ごとに状況変わります)
段数が増えるにつれ、任意のステップ量だけ数値を減らす方向。

①arduino IDEのインストール

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

<https://www.arduino.cc/en/Main/Software>

からIDE(開発環境)をDL、インストールする。

②ライブラリの追加

標準装備外のライブラリを追加する必要があります。

programfiles>Arduino>librariesの下に各ライブラリ名のフォルダを作って

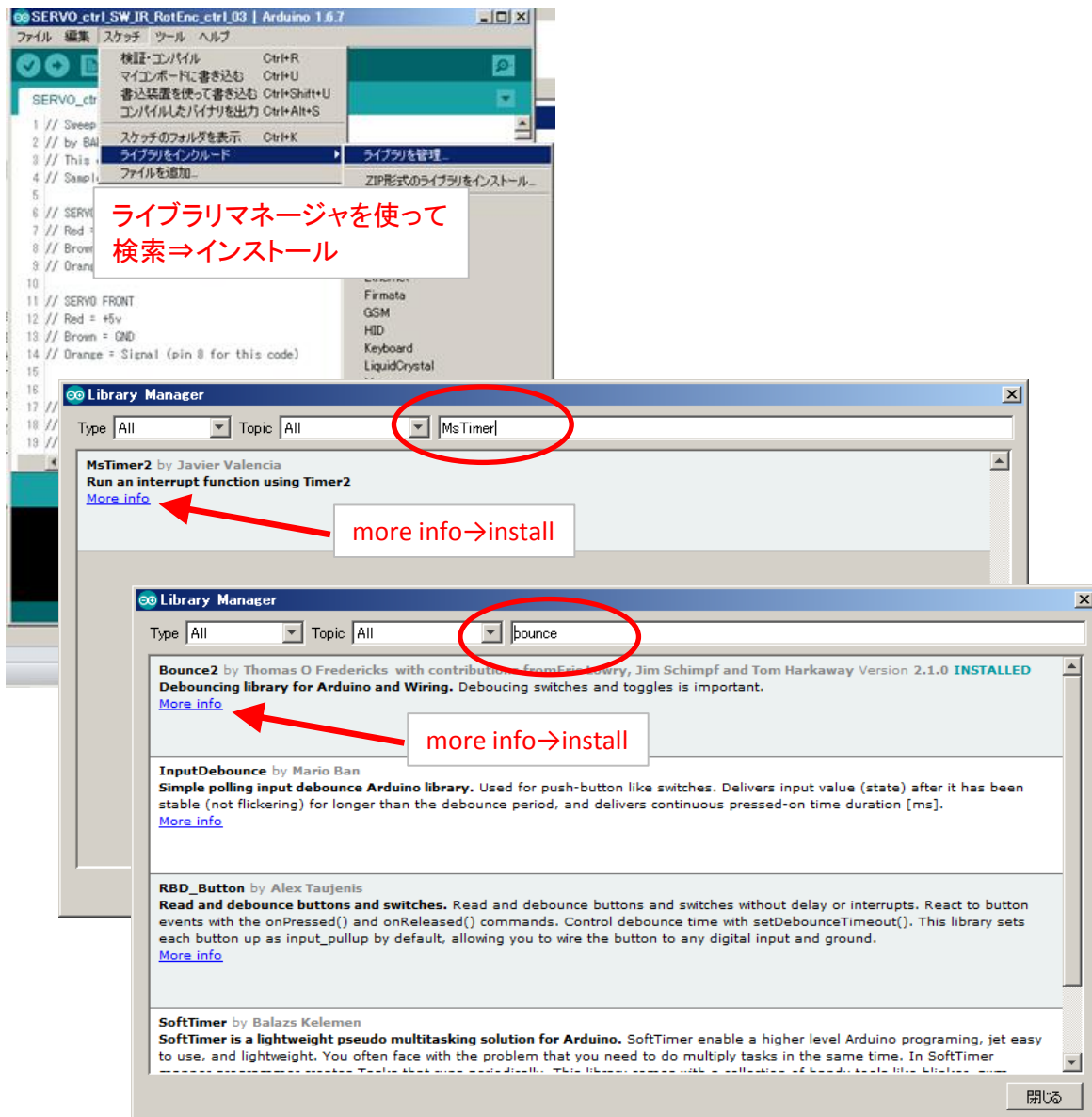
ライブラリのzip内容を展開しておく。

・MsTimer2: タイマーライブラリ

<http://www.musashinodenpa.com/arduino/ref/index.php?f=1&pos=2021>

・Bounce2: チャタリング対策ライブラリ

<http://playground.arduino.cc/Code/Bounce>



③arduinoMCUボードの導入

③-1 promicroの場合

<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/installing-windows>

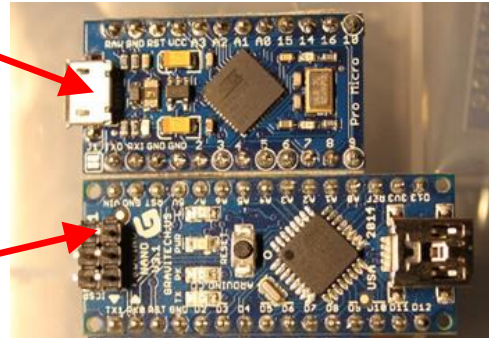
ドライバをDLLデスクトップ等に展開しておく。

MCUボードをPCにUSB接続する。

上記ドライバでセッティングする。

“ボードマネージャ”でMCU種を指定し、設定する。上記URL中段からの説明参照。

注: MCU設定は5V16MHzを選ぶこと。



③-2 nano v3(モドキ) の場合

<https://www.arduino.cc/en/Main/ArduinoBoardNano>

<http://www.dorapro.co.jp/engineerblog/?p=720>

CH340ドライバをDLLデスクトップに展開しておく。

MCUボードをPCにUSB接続する。

上記ドライバでセッティングする。

“ボードマネージャ”でMCU種を指定し、設定する

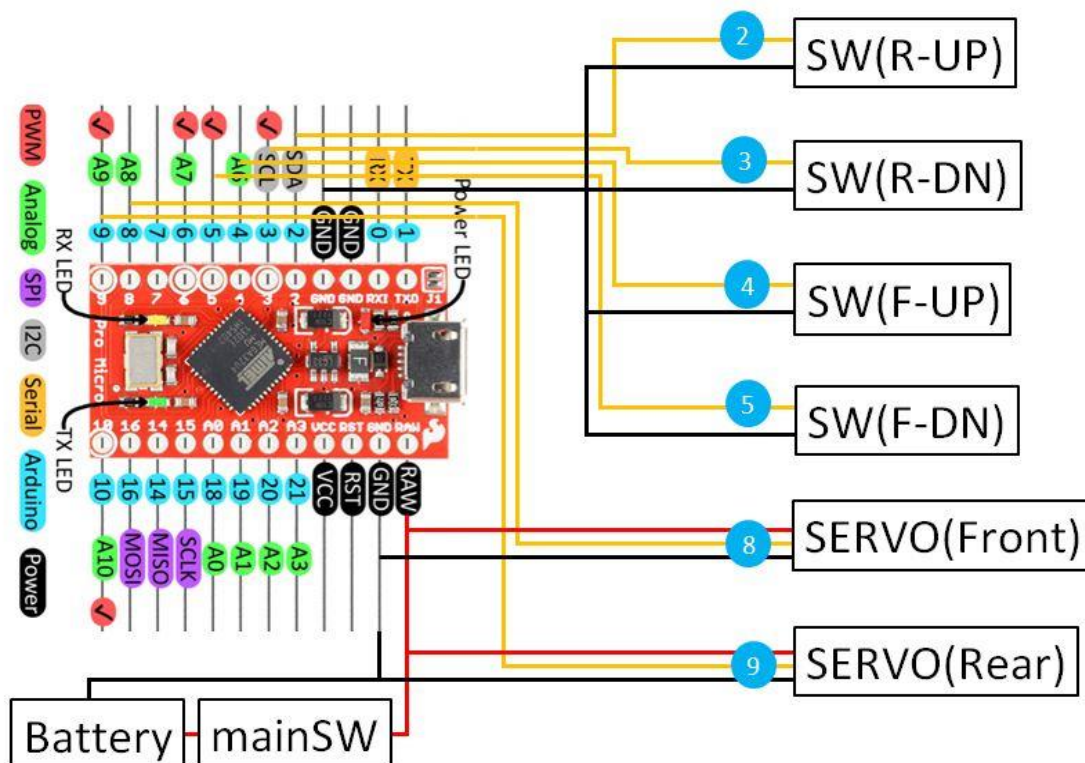
Select “Arduino Diecimila, Duemilanove, or Nano w/ ATmega168” or

“Arduino Duemilanove or Nano w/ ATmega328”

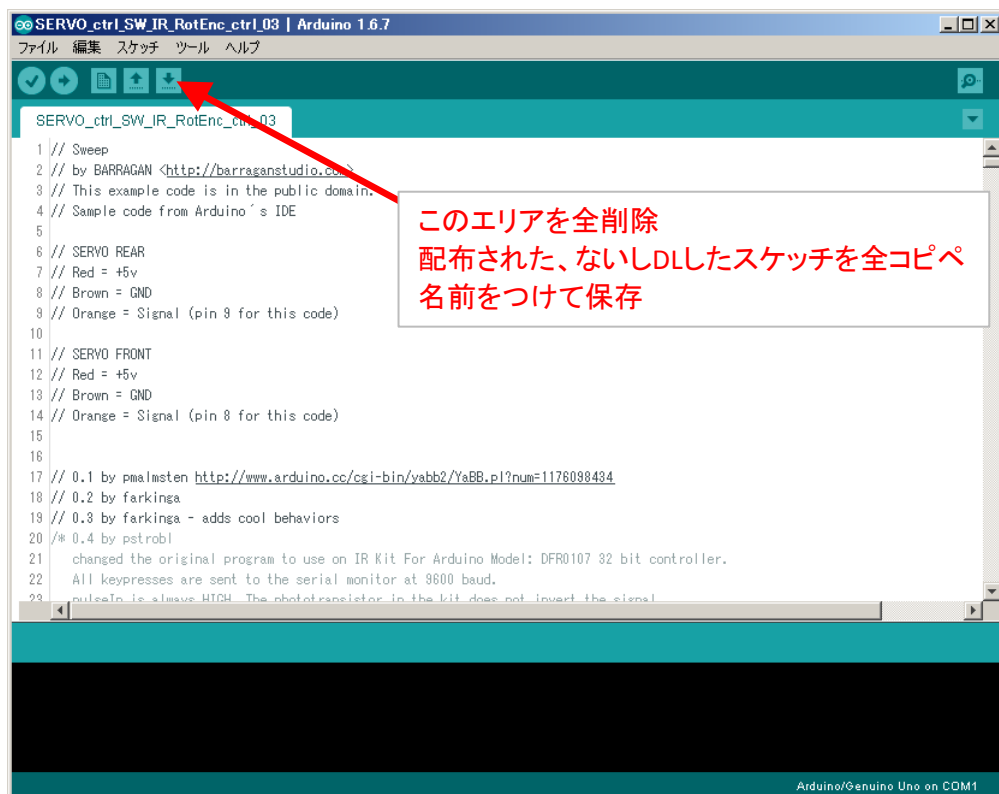
from the Tools > Board menu (according to the microcontroller on your board).

注: MCU設定は5V16MHzを選ぶこと。

④system



⑤スケッチの準備・編集



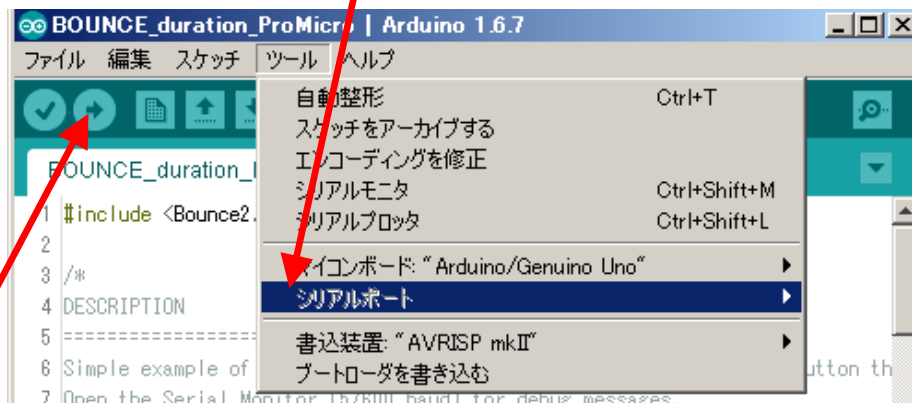
⑥スケッチのMCUへの書き込み

※スケッチの書き込みはMCUを単体にしてPCと接続した方が安定して動作します。

⑥-1 初期設定の書き込み(書き込み履歴のないMCUで最初の1回目だけ必要な作業。)

xxxx.inoを開く

comポートが開かれていることを確認する。



⑥-2 メインスケッチの書き込み

xxxx.inoを開く

上記同様、comポートが開かれていることを確認する。

設定パラメータ確認後、スケッチの書き込みを実施する。

※シフトセッティング等をチューニングするときは、都度パラメータを更新したスケッチを書き込んでください。

⑦メインスケッチパラメータ説明

スケッチ中に下記パラメータ群があります。セッティング後、MCUに書き込んで反映させる。

pin assignment

```
35 #define BUTTON_PIN 2 //REAR SWITCH (UP)
36 #define BUTTON_PIN2 3 //REAR SWITCH (DN)
37 #define BUTTON_PIN_f1 4 //FRONT SWITCH (UP)
38 #define BUTTON_PIN_f2 5 //FRONT SWITCH (DN)
39 #define SERVO_F 8 //FRONT SERVO
40 #define SERVO_R 9 //REAR SERVO
41 #define LED_PIN2 10
42
```

MCUと周辺回路の接続を変える場合は
ピンアサインのパラメータをいじる。

RearSetting(T.B.D)

```
96 //REAR SHIFT SETTING PARAMETER START
97
98 int pos_zero0 = 2050; // shift 0 position 180deg ->1st gear direction, 0deg ->10th gear direction,
99 int pos_zero = 2050; // shift 0 position 180deg ->1st gear direction, 0deg ->10th gear direction,
100 int pos_shiftstep = 56; // shift step degree 68->56 @150809
101 int pos_data[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
102
103 int pos_shiftover_up = 40; // shift over degree
104 int pos_shiftover_down = 40; // shift over degree
105 int ovsft_delay = 500; // over shift delay [msec]
106 int sft_delay = 20; // shift delay per gear pos [msec]
107
108 //REAR SHIFT SETTING PARAMETER END
```

pos_zero = 1st位置の決定
pos_shiftstep = 1段ごとのピッチ
(数値はdegではないです)

pos_data= 各ギヤごとのfine tune data
上記パラメータで追い込みきれないときに使用

FrontSetting(T.B.D)

```
116 //FRONT SHIFT SETTING PARAMETER START
117
118 int pos_zero_f = 80; // shift 0 position 180deg ->1st gear direction, 0deg ->10th gear direction,
119 int pos_inner = 70; //80->70 @150809
120 int pos_outer = 150; //120->120 @150809
121 int pos_shiftstep_f = 5; // shift step degree
122 int pos_shiftover_up_f = 30; // shift over degree 15->20 @150809
123 int pos_shiftover_down_f = 30; // shift over degree 15->40 @150809
124 int ovsft_delay_f = 2000; // over shift delay [msec]
125 int sft_delay_f = 20; // shift delay per gear pos [msec]
126
127 //FRONT SHIFT SETTING PARAMETER END
```

⑧動作確認

メカ操作中にUSBを接続し、シリアルポート開通を確認ののち、シリアルモニタを開くと、メカ操作時ごとの動作パラメータが表示されます。デバッグ用にどうぞ。
(デバッグ時はシステムから切り離す必要はないが、書き込みは単体でやった方が安定動作しま

