

C-Style for C-cubic

操作ガイド

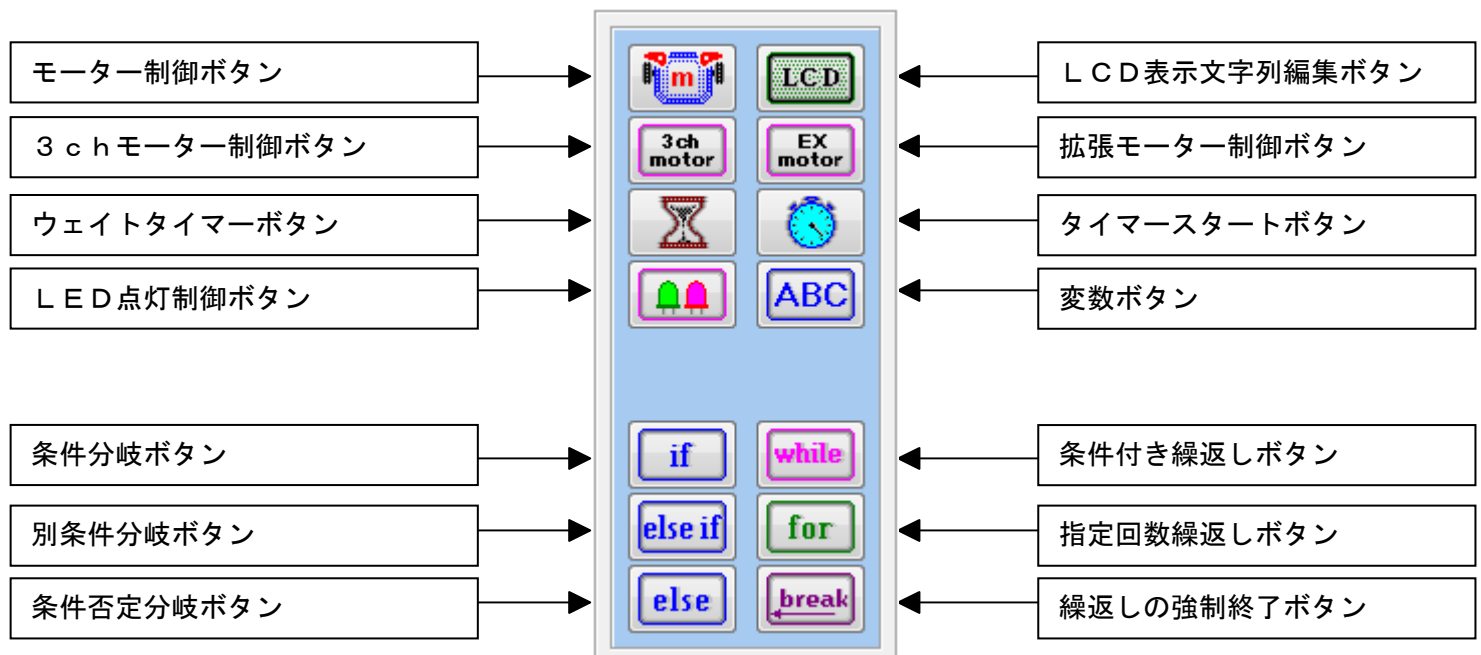
目 次

1	プログラムボタンの説明	2
1.1	プログラムボタンリスト	2
1.2	モーター制御ボタン	3
1.3	3ch モーター制御ボタン	4
1.4	拡張モーター制御ボタン	5
1.5	ウェイトタイマーボタン	6
1.6	LED 制御ボタン	7
1.7	LCD 表示文字列の編集ボタン	8
1.8	タイマースタートボタン	9
1.9	変数ボタン	10
1.10	条件分岐	11
1.11	条件付き繰返し	12
1.12	回数指定の繰返し	13
1.13	タイマーチェック	14
1.14	変数チェック	15
2	プログラムボタンの挿入と削除	16
2.1	ボタンの挿入	16
2.2	ボタンの削除	17
2.3	ボタンのコピーと貼付け	18
3	入出力設定と拡張機能の設定	19
3.1	Setup ボタンの表示	19
3.2	入出力設定	20
3.3	拡張機能の設定	21
3.4	超音波距離センサーを使う	22
3.5	サーボモーターを使う	24
3.6	6D/9D-Compass（多機能電子コンパス：DSR1005/DSR1603）を使う	27
3.7	複数台のガジェット を接続する	29
3.8	EG シリーズを使う	30
3.9	Pixy カメラを使う	30
4	サブプログラムの説明	31
4.1	サブプログラムボタンを置く	31
4.2	サブプログラムの編集	32

1 プログラムボタンの説明

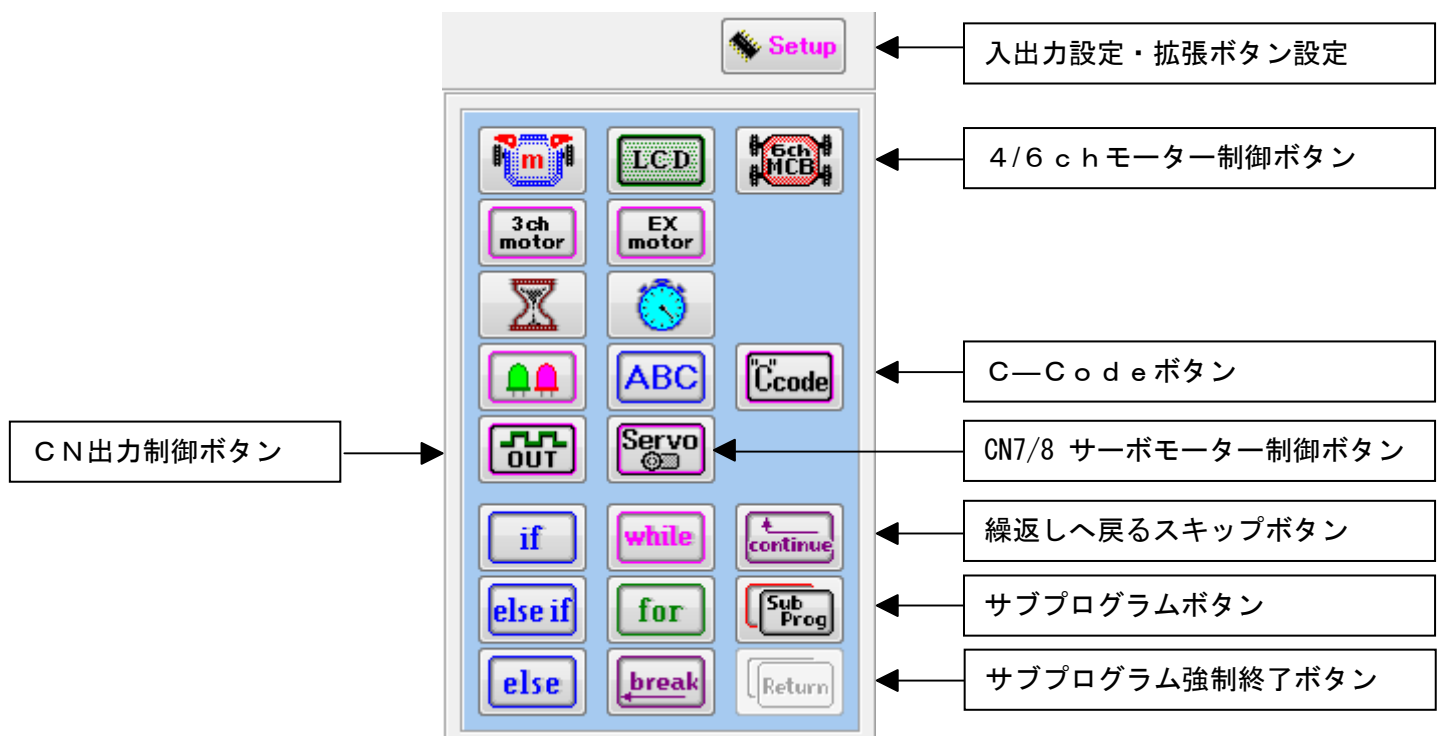
1.1 プログラムボタンリスト

標準のプログラムボタンリスト



◆拡張機能を設定するとプログラムボタンリストも下記のように変わります。

拡張機能設定時のプログラムボタンリスト

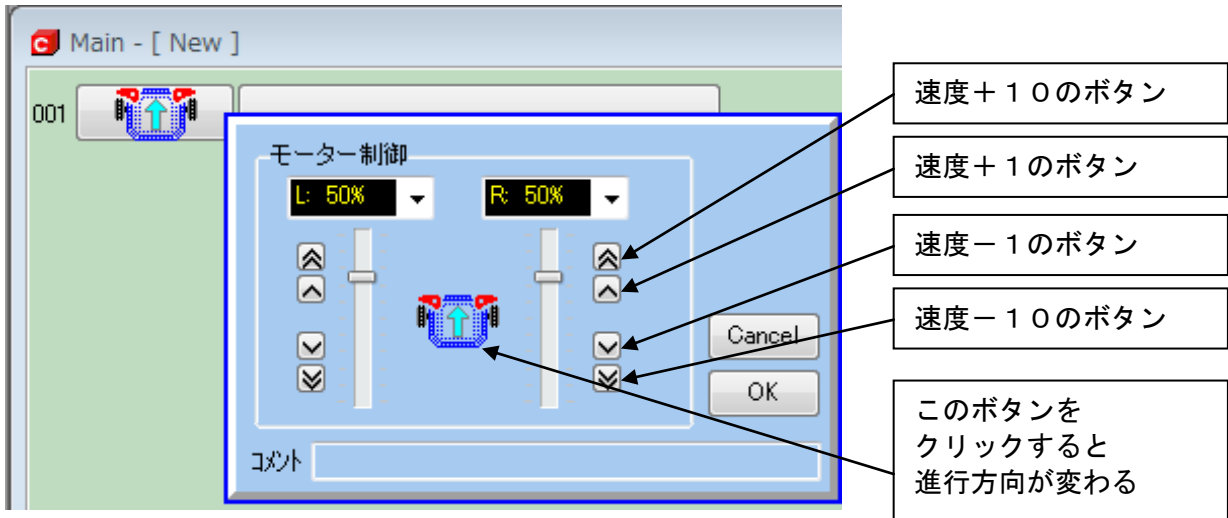


詳しくは、「3 入出力設定と拡張機能の設定」のページを参照してください。

1.2 モーター制御ボタン



回転速度を設定してロボットの進む方向を決めます。（L:左側、R:右側）



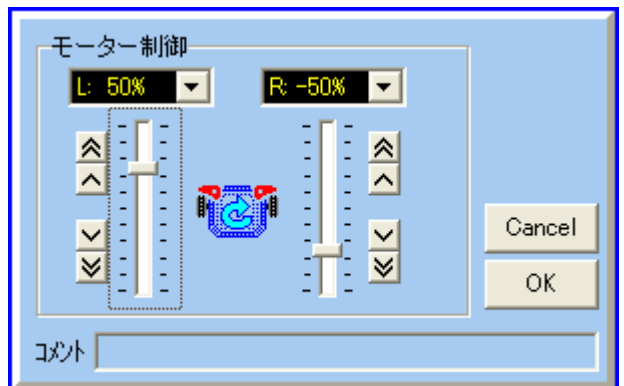
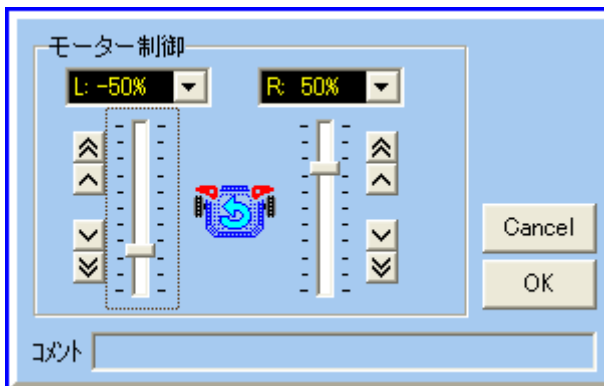
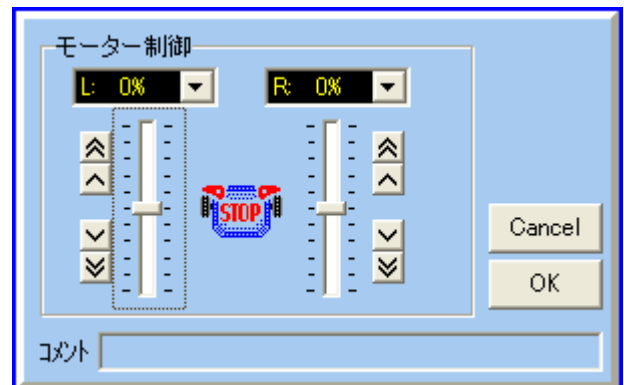
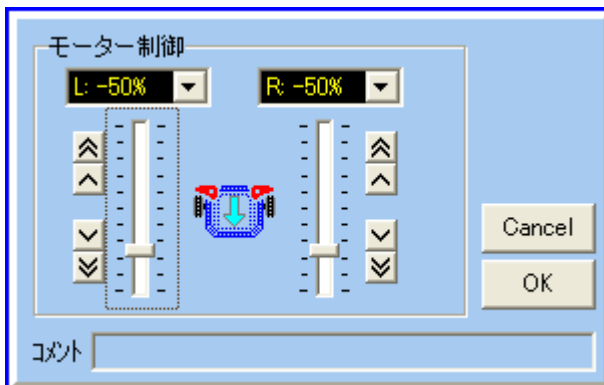
1～100 が前回転速度です。

-1～-100 が後回転速度です。

0 はモーターが停止します。

左右を+/-逆に設定するとロボットは回転します。

中央のモーターボタンをクリックすると前進、後退、停止、左回転、右回転の順に進行方向を簡単に設定することができます。左右どちらかの回転速度を設定してから中央のモーターボタンをクリックすると、反対側のモーターの回転速度も同じ値に設定されます。

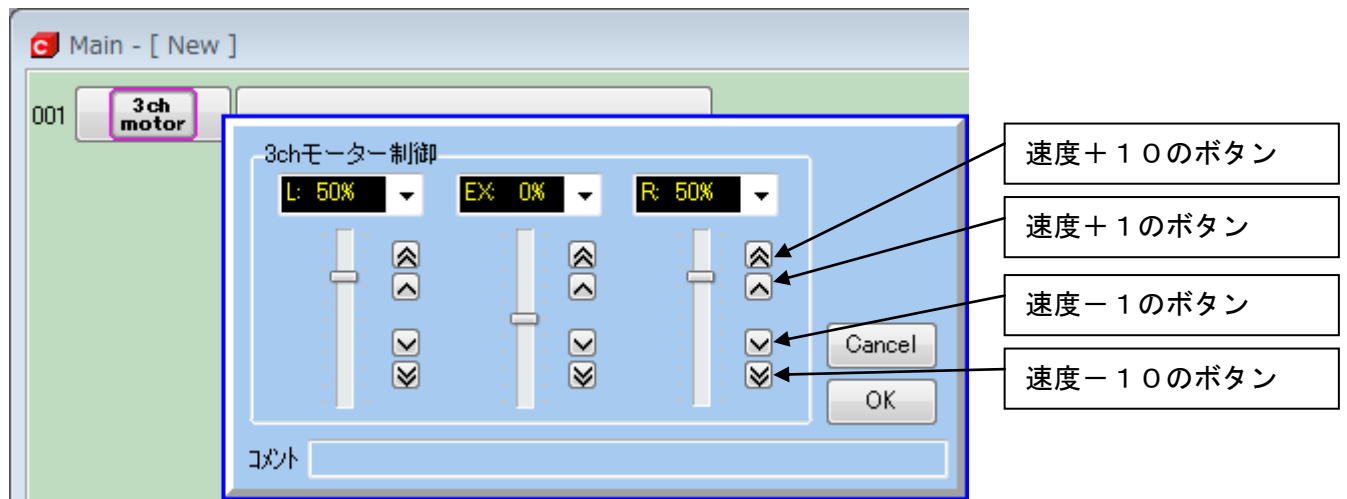


前方向に旋回または、後方向に旋回にするには、速度ボタンまたはスライダーを操作して決めます。速度表示のプルダウンメニューを表示して変数 A～P で指定することもできます。

1.3 3ch モーター制御ボタン

3ch
motor

L モーターと R モーターの回転速度を設定してロボットの進む方向を決めます。（L: 左側、R: 右側）
EX モーターの回転速度を設定することができます。（EX: 拡張）



1～100 が前回転速度です。

-1～-100 が後回転速度です。

0 はモーターが停止します。

左右を+/-逆に設定するとロボットは回転します。

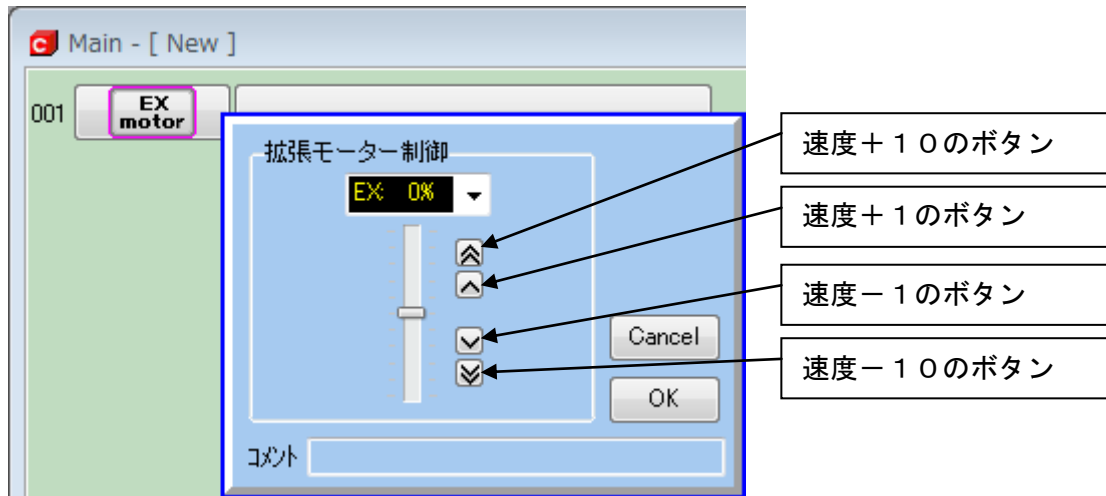
前進、後退、停止、左回転、右回転、前方向に旋回または、後方向に旋回にするには、左右の速度ボタンまたはスライダーを操作して決めます。

速度表示のプルダウンメニューを表示して変数 A～P で指定することもできます。

1.4 拡張モーター制御ボタン



EX モーターの回転速度を設定することができます。（EX: 拡張）



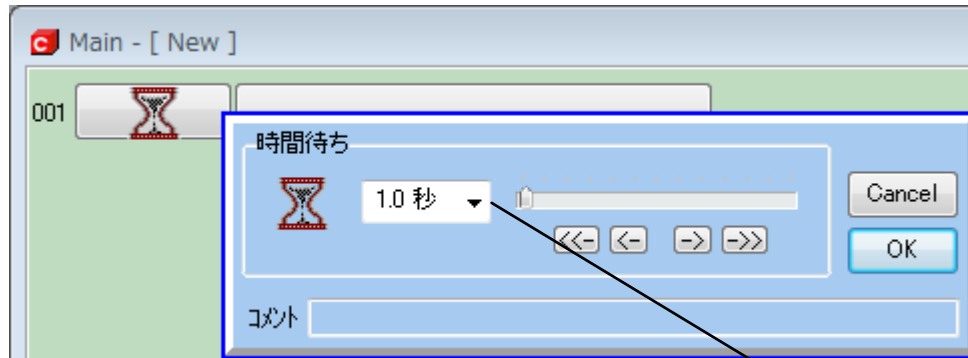
1～100 が前回転速度です。

-1～-100 が後回転速度です。

0 はモーターが停止します。

速度表示のプルダウンメニューを表示して変数 A～P で指定することもできます。

1.5 ウェイトタイマーボタン

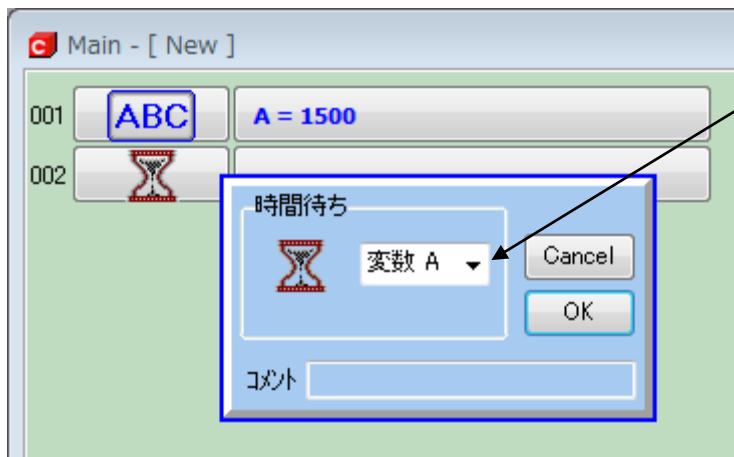
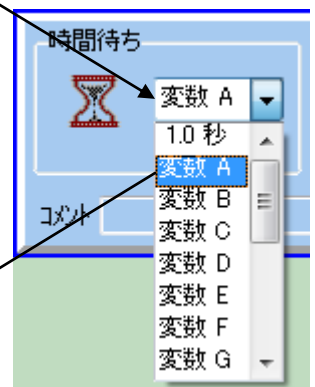


0.1 秒から 60.0 秒までの時間待ちを設定します。

待ち時間を変数に設定することもできます。

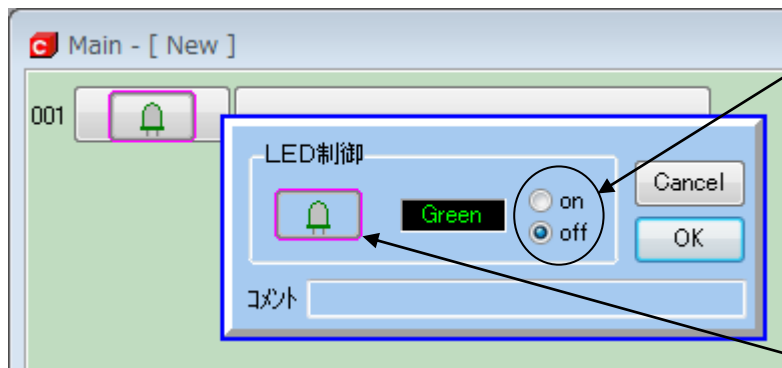
変数へ待ち時間を設定する場合は、1000 倍したミリ秒の単位で設定します。

例えば、1.5 秒の待ち時間を変数へ設定する場合は、A=1500 として待ち時間表示をクリックして「変数 A」を選択します。



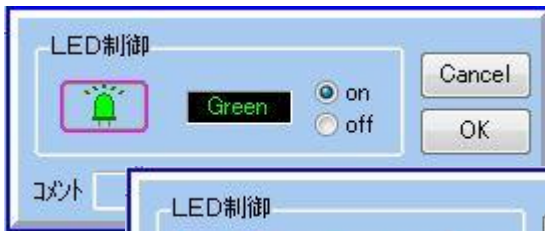
ウェイトタイマーボタンは、簡単に待ち時間をプログラムすることができますが、待っている間は、他のプログラムボタンを置いて制御することができません。待っている間他のプログラム制御したい場合は、タイムスタートボタンとタイムチェックを組み合わせで行います。

1.6 LED 制御ボタン

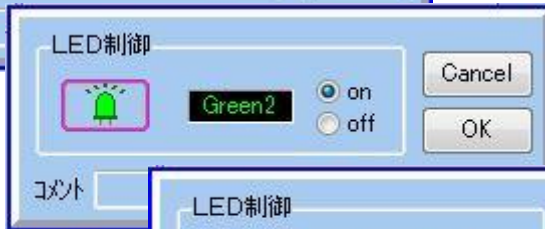


ここをクリックして点灯、消灯を選択します。

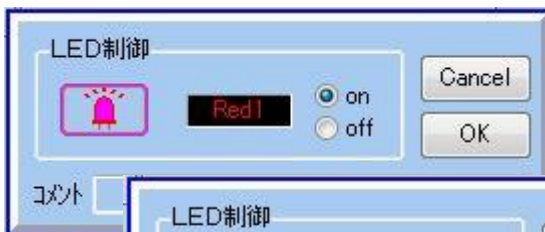
ここをクリックして赤と緑のLEDを選択します。



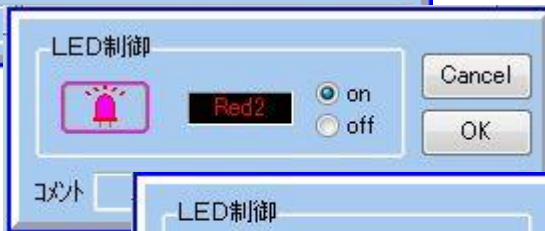
緑LEDの点灯



緑LED 2～3の点灯

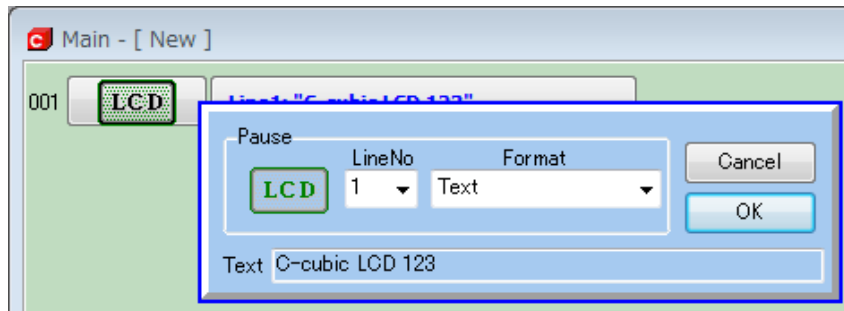


赤LED 1～3の点灯



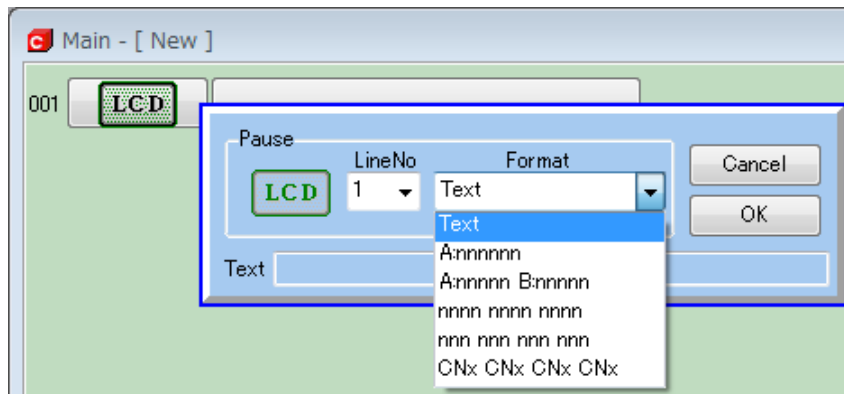
赤LED 1の消灯

1.7 LCD 表示文字列の編集ボタン



2 行 × 16 桁の LCD への文字列は、行番号を指定して編集することができます。

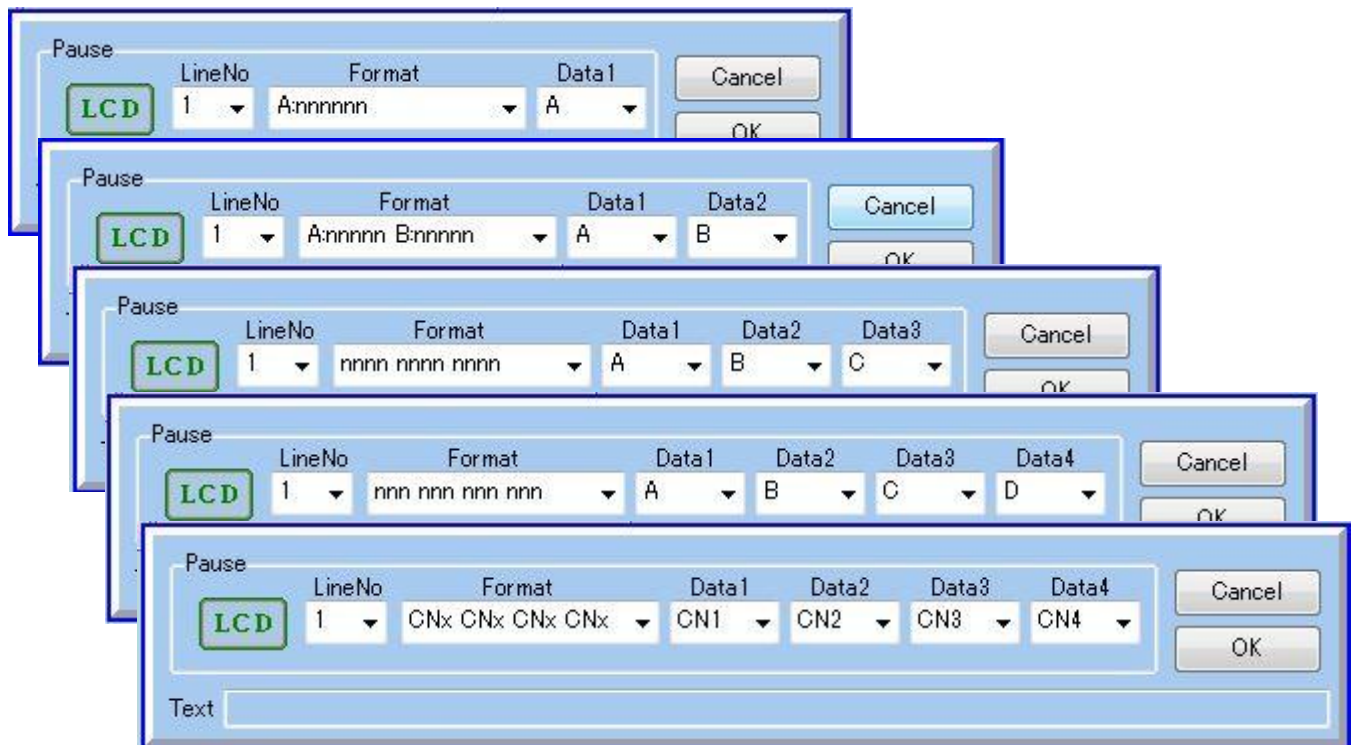
行番号 (Line No) は 1、または 2 を選択します。



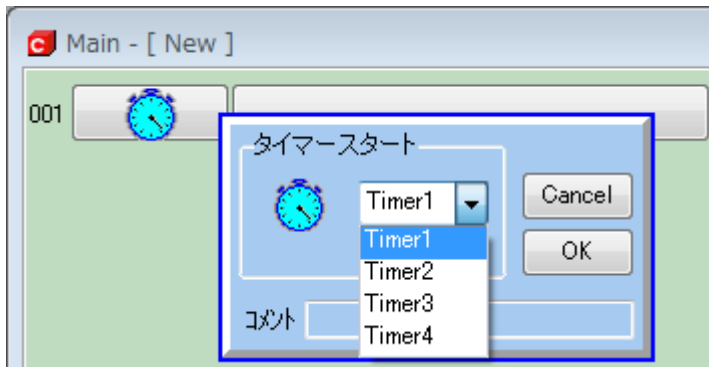
Format 欄の「Text」では、下の Text 欄に表示したい文字列を入力します。

半角 16 文字までの英数字と記号が入力できます。

但し、ダブルコーテーション「”」の記号は入力できません。(自動的に付加されます)
その他の表示フォーマットを選択すると、変数値や CN のセンサー値を表示できます。



1.8 タイマースタートボタン

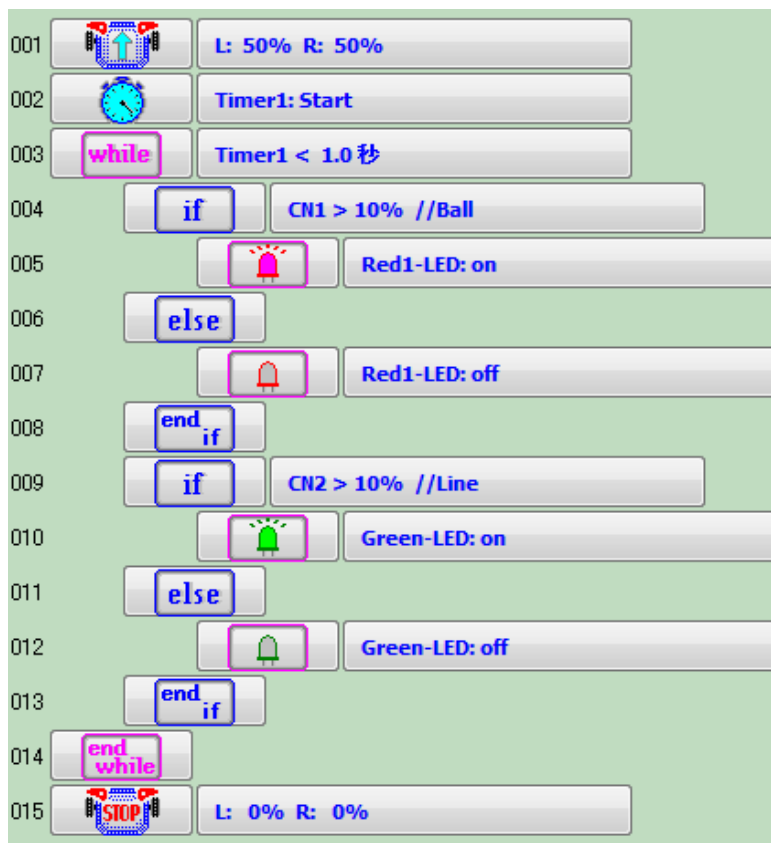


条件判定のタイムチェックで使用するタイマーをスタートさせます。

4 個のタイマーを選択して使用します。

ストップウォッチのスタートボタンを押す感じと同じですが、停止はありません。

タイマーを使ったプログラム例



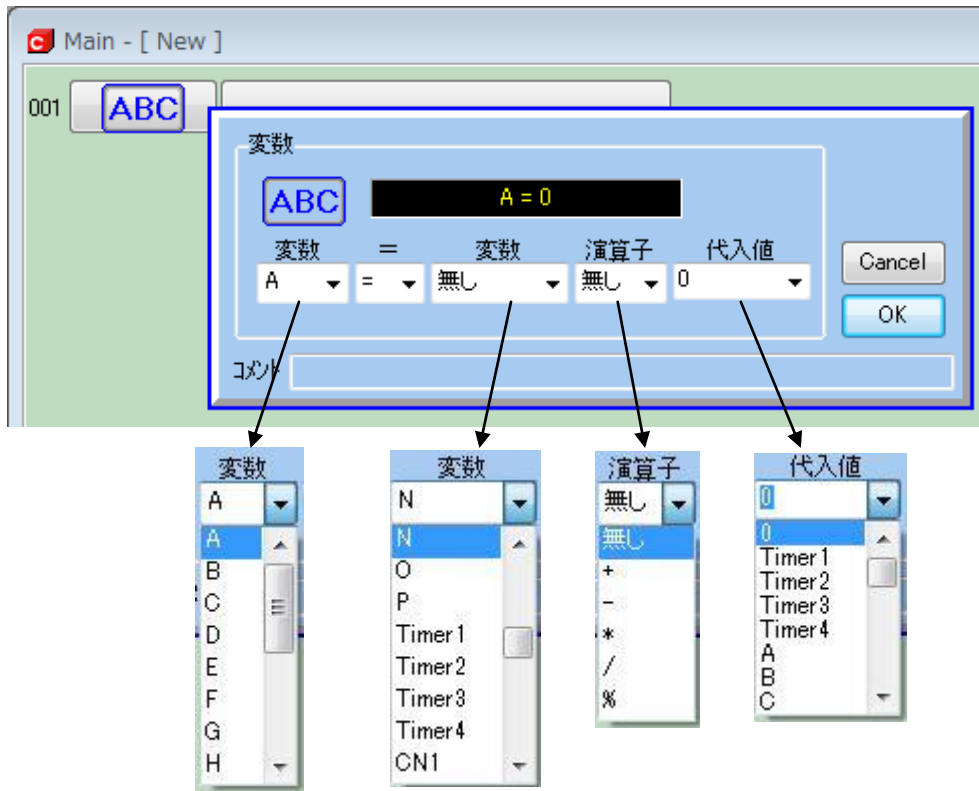
この例は、C-Style for C-cubic 導入編の「3. ロボットを動作させるまでの手順」で説明した 1 秒間前進して停止させるプログラムを タイマーを使って同じ動作を実現しています。

ウェイトタイマーを使用した場合、ただ 1.0 秒間待つだけで、他のプログラムを実行することができませんでした。タイマーを使った場合 1.0 秒間に他のプログラムを実行することができます。

この例では、1.0 秒間の前進の間に CN1 のボールセンサーが 10%以上の判定で RED1 の LED を点灯し、CN2 のラインセンサーが 10%以上の判定で緑 LED の点灯することができます。

1.9 変数ボタン

ABC

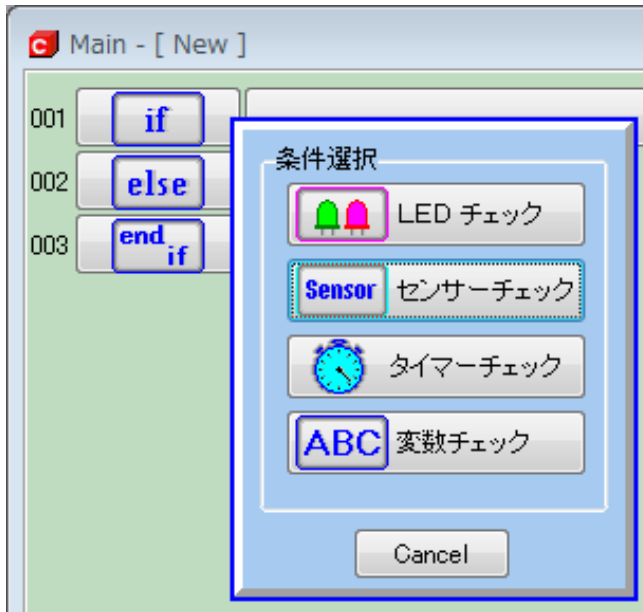


変数は、A～P の 16 個が使用できます。
演算子は、+、-、×、÷、余り計算が使用できます。またタイマーの値や CN1 から CN20 に接続されたセンサー値も変数へ代入することができます。

◇変数への代入例

"A = 0"	変数 A に 0 を代入
"A = 100"	変数 A に 100 を代入
"B = A"	変数 B に変数 A を代入
"A = Timer1"	変数 A にタイマー 1 の値を代入 (変数 A は 0~65535 の値になる)
"A = A + 1"	変数 A に変数 A の値に 1 を足した値を代入 (変数 A は以前の値より 1 増す)
"A = A * 2"	変数 A に変数 A を 2 倍にした値を代入 (変数 A は以前の値の倍になる)
"A = A / 2"	変数 A に変数 A を 2 で割った値を代入 (変数 A は以前の値の半分になる)
"A = B % 2"	変数 A に変数 B を 2 で割った余りの値を代入 (変数 A は 0 または 1 となる)
"A = B - CN6"	変数 A に変数 B から CN6 に接続されたセンサー値を引いた値を代入

1.10 条件分岐



if イフと呼びます。（条件成立で次の行からプログラムを実行します。）

条件成立で、この間に置かれたプログラムボタンが実行されます。

else if エルス イフと呼びます。（条件成立で次の行からプログラムを実行します。）

条件成立で、この間に置かれたプログラムボタンが実行されます。

else エルスと呼びます。（条件不正立の場合に次の行からプログラムを実行します。）

どの条件も成立しなかった時、この間に置かれたプログラムボタンが実行されます。

end if エンド イフと呼びます。

「if」と「end if」の組合せで必ず配置されます。

「if」と「else if」ボタンを置いた時に、ボールセンサー、ラインセンサー、タッチセンサー、タイムチェック、変数チェックの判定条件を選択することができます。

「else if」は「if」ボタンの後ろに幾つでも置けます。

「else」ボタンは「end if」ボタンの前に1個だけ置けます。

条件が成立した場合は、次の行からプログラムは実行され、「else if」または「else」ボタンに出会うと「end if」までスキップします。

条件が不正立の場合は、次の「else if」か「else」までスキップし、無ければ「end if」までプログラムはスキップします。

条件分岐ボタンの間にその他のボタン（さらに条件分岐ボタンでも良い）を挿入するには、挿入したいプログラムボタンをボタンリストから選択しておいてから、挿入したい行のプログラムボタンをクリックすると挿入されます。

例えば、1行目と2行目の間にプログラムボタンを挿入する場合は2行目のボタンをクリックします。

1.11 条件付き繰り返し

**while**

ホワイルと呼びます。

条件成立の間、この間に置かれたプログラムボタンが繰り返し実行されます。

end while

エンド ホワイルと呼びます。

「while」と「end while」ボタンの組合せで必ず配置されます。

別のプログラムボタンを挿入する場合は、先にプログラムボタンリストから挿入したボタンを選択しておいてから、挿入したい行のボタンをクリックします。(条件分岐の場合と同じ)

break

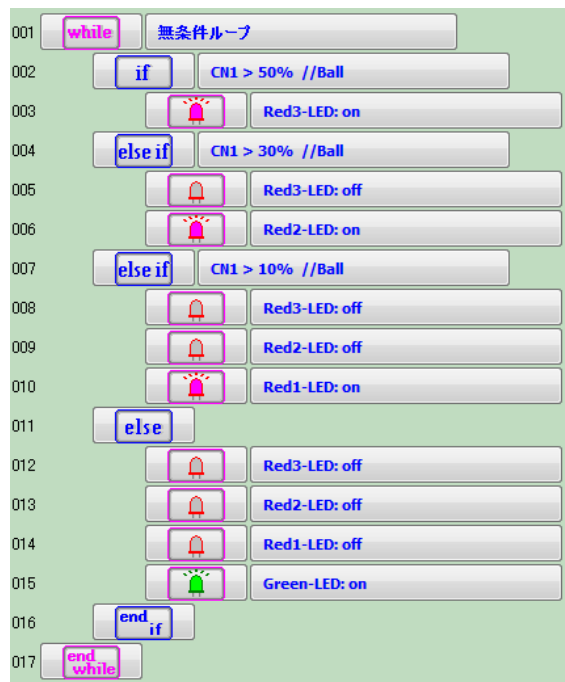
ブレイクと呼びます。繰り返しの中にだけ置けるボタンで、このボタンに出会うと、繰り返し条件を無視して強制終了します。(「end while」までのプログラムボタンは実行されません)

continue

コンティニューと呼びます。繰り返しの中にだけ置けるボタンでこのボタンに出会うと、繰り返しの条件判定に戻ります。(「end while」ボタンまでのプログラムボタンは実行されません)

このボタンは拡張プログラムボタンの説明で“Advanced Mode”の設定で表示されます。

while, if, else if, else プログラム例



while(001)～end while(017)の範囲を無条件で繰り返す

002～003: CN1 のボールセンサーが 50%以上で

RED3 の LED を点灯

004～006: CN1 のボールセンサーが 30%以上で

RED3 の LED を消灯

RED2 の LED を点灯

007～010: CN1 のボールセンサーが 10%以上で

RED3, RED2 の LED を消灯

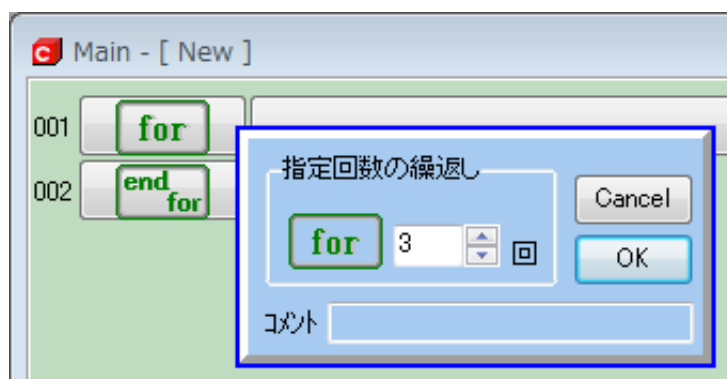
RED1 の LED を点灯

011～015: 上記の条件以外の場合は

RED3, RED2, RED1 の LED を消灯

緑 LED を点灯

1.12 回数指定の繰り返し



for フォアと呼びます。

指定回数、この間に置かれたプログラムボタンを実行します。

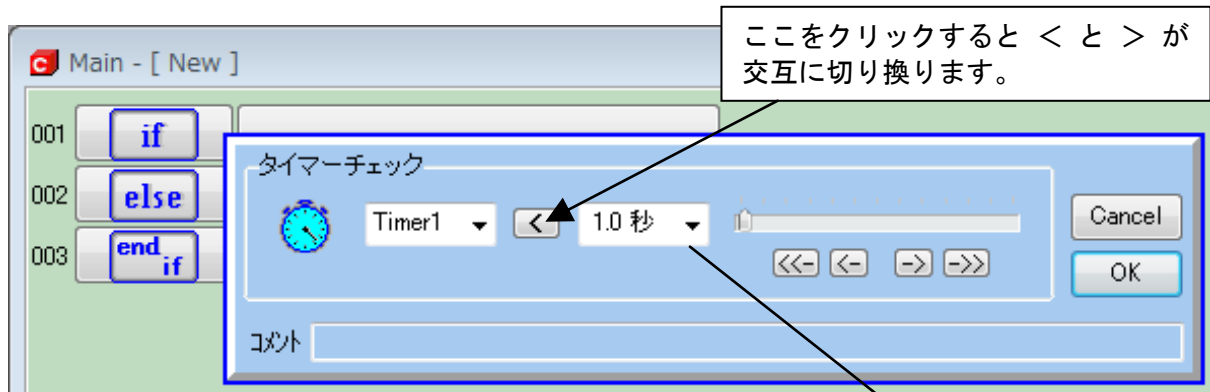
end for エンド フォアと呼びます。

別のプログラムボタンを挿入する場合は、先にプログラムボタンリストから挿入したボタンを選択しておいてから、挿入したい行のボタンをクリックします。（条件分岐の場合と同じ）

break ブレイクと呼びます。 繰り返しの中にだけ置けるボタンで、このボタンに出会うと、繰り返し回数を無視して強制終了します。（「end for」までのプログラムボタンは実行されません）

continue コンティニューと呼びます。 繰り返しの中にだけ置けるボタンでこのボタンに出会うと、繰り返し回数の判定に戻ります。（「end for」ボタンまでのプログラムボタンは実行されません）
このボタンは拡張プログラムボタンの説明で“Advanced Mode”の設定で表示されます。

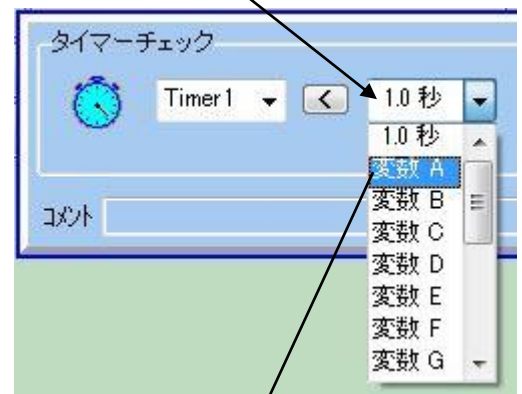
1.13 タイマーチェック



4 個のタイマーを選択して、設定値と比較することで条件判定することができます。
設定値のかわりに、変数（A～P）との比較判定もできます。

ボタン名称の例

- “Timer1 < 0.1 秒 より小さい”
- “Timer2 > 1.5 秒 より大きい”
- “Timer1 > 変数 A より大きい”
- “Timer2 < 変数 A より少ない”



◇変数と比較する場合の注意

変数へは秒単位を 1000 倍したミリ秒の
単位で値を代入する必要があります。
例えば 1.5 秒の値を変数に代入して比較したい時
は、1000 倍して 1500 を代入します。

変数 A = 1500

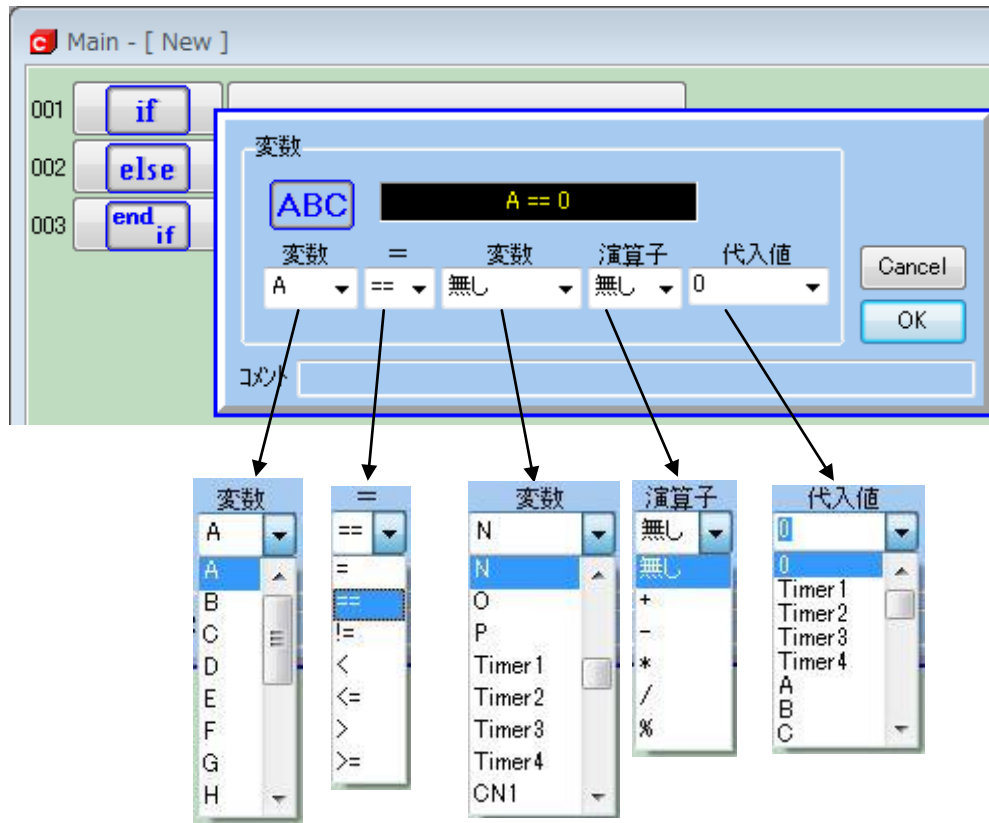
```
if Timer1 < 変数A
```

```
...
```

```
endif
```



1.14 変数チェック



変数は、A～Pの16個が使用出来ます。

演算子と＋、－、×、÷、余り計算などが行えます。またタイマーの値やCN1からCN20に接続されたセンサー値も変数へ代入することが出来ます。

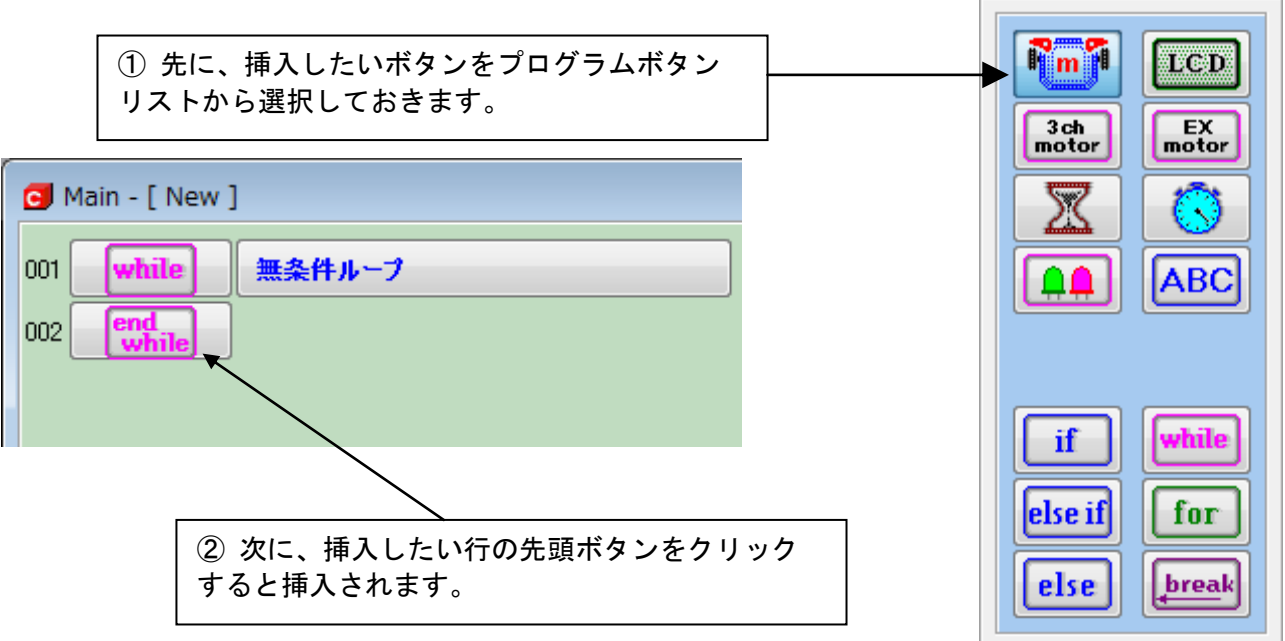
ボタン名称の例

“A == 0”	変数 A が 0 の場合
“A < 5”	変数 A が 5 より小さい場合
“A > B”	変数 A が変数 B より大きい場合
“A < CN1”	変数 A より CN1 のセンサー値の方が大きい場合
“A < B - CN2”	変数 A より変数 B から CN2 のセンサー値を引いた値の方が大きい場合
“A == B % 2”	変数 A が変数 B を 2 で割った余りと等しい場合（変数 A が奇数か？の判定）
“A > CN3”	変数 A より CN3 のセンサー値の方が小さい場合
“A < Timer3”	変数 A よりタイマー3の値の方が大きい場合


2 プログラムボタンの挿入と削除

2.1 ボタンの挿入

① 先に、挿入したいボタンをプログラムボタンリストから選択しておきます。

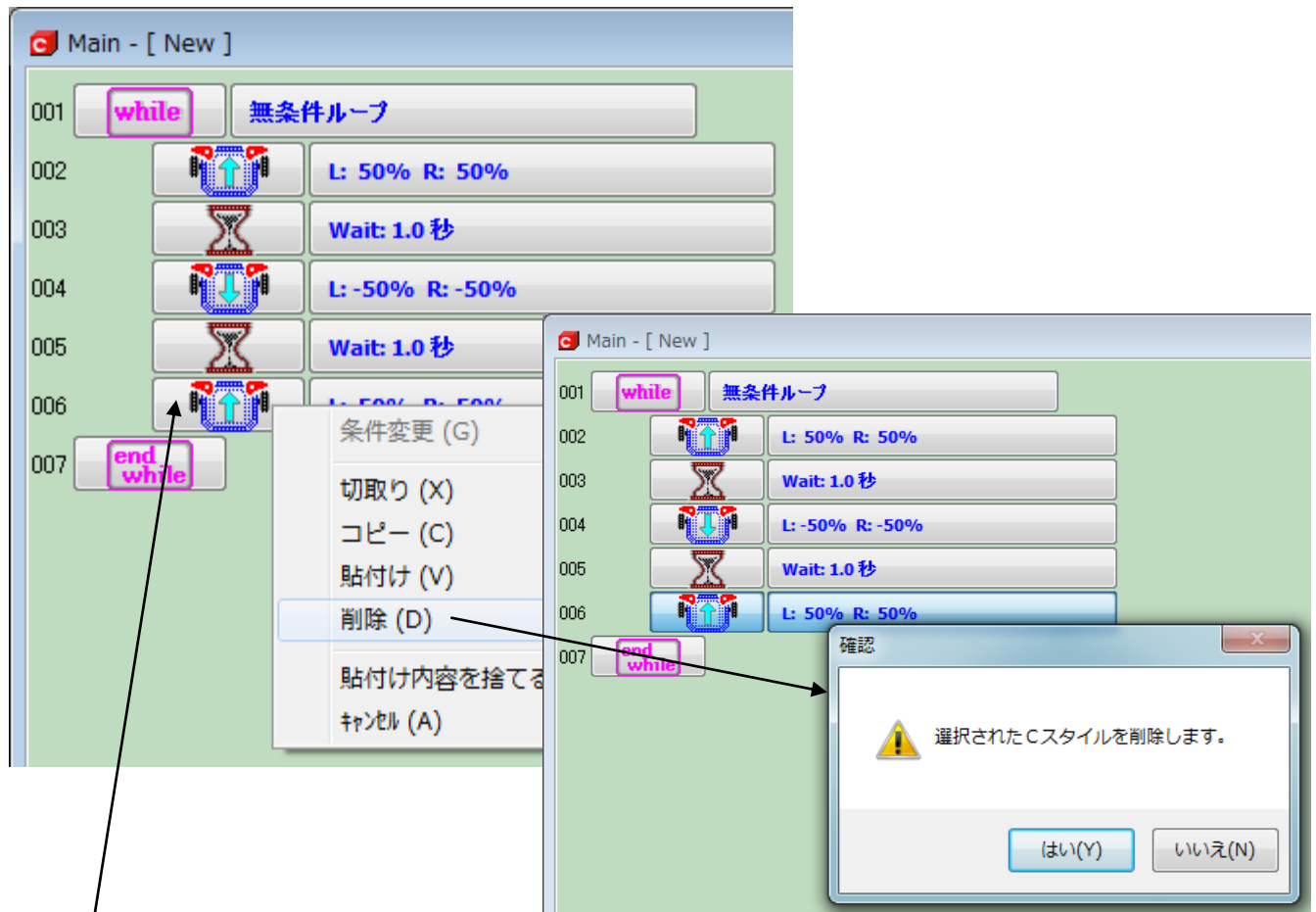


② 次に、挿入したい行の先頭ボタンをクリックすると挿入されます。



この例では、モーターボタンをプログラムボタンリストから選択してから、「end while」ボタンをクリックすると前進ボタンが挿入されたところです。

2.2 ボタンの削除



- ① 削除したい行の前側のボタンをクリックするとポップアップメニューが表示されますので、「削除」を選択します。
- ② ボタン削除の確認ダイアログが表示されますので、「はい」をクリックすると、その行のボタンが削除され、行が詰められます。

ポップアップメニューには、「切取り」、「コピー」、「貼付け」などの機能もあります。

「while」や「for」または「if」ボタンを選択した場合は、「end while」や「end for」または「end if」までの範囲が削除などの対象となります。

2.3 ボタンのコピーと貼付け

コピーしたいボタンの先頭をクリックしてポップアップメニューのコピーを選択する。

貼付けたいボタンの先頭をクリックしてポップアップメニューの貼付けを選択するとコピーしたボタンが貼付け（挿入）されます。

コピーを連続して行くと、以前にコピーした内容の後に追加され、貼付け時に全て挿入されます。

この機能は、切取りにおいても同じ働きをします。

コピーや切取りで蓄積された貼付け内容がなくなった場合は、ポップアップメニューを表示させて、「貼付け内容を捨てる」を選択します。

3 入出力設定と拡張機能の設定

C-cubic には、CN1 から CN20 までの入出力端子があります。

出荷時の状態では、CN1～CN4 は赤外線ボールセンサー、ラインセンサー、左右タッチセンサーに接続され、CN5～CN8、CN11～CN20 までは予備で、これらはアナログ入力の設定となっています。

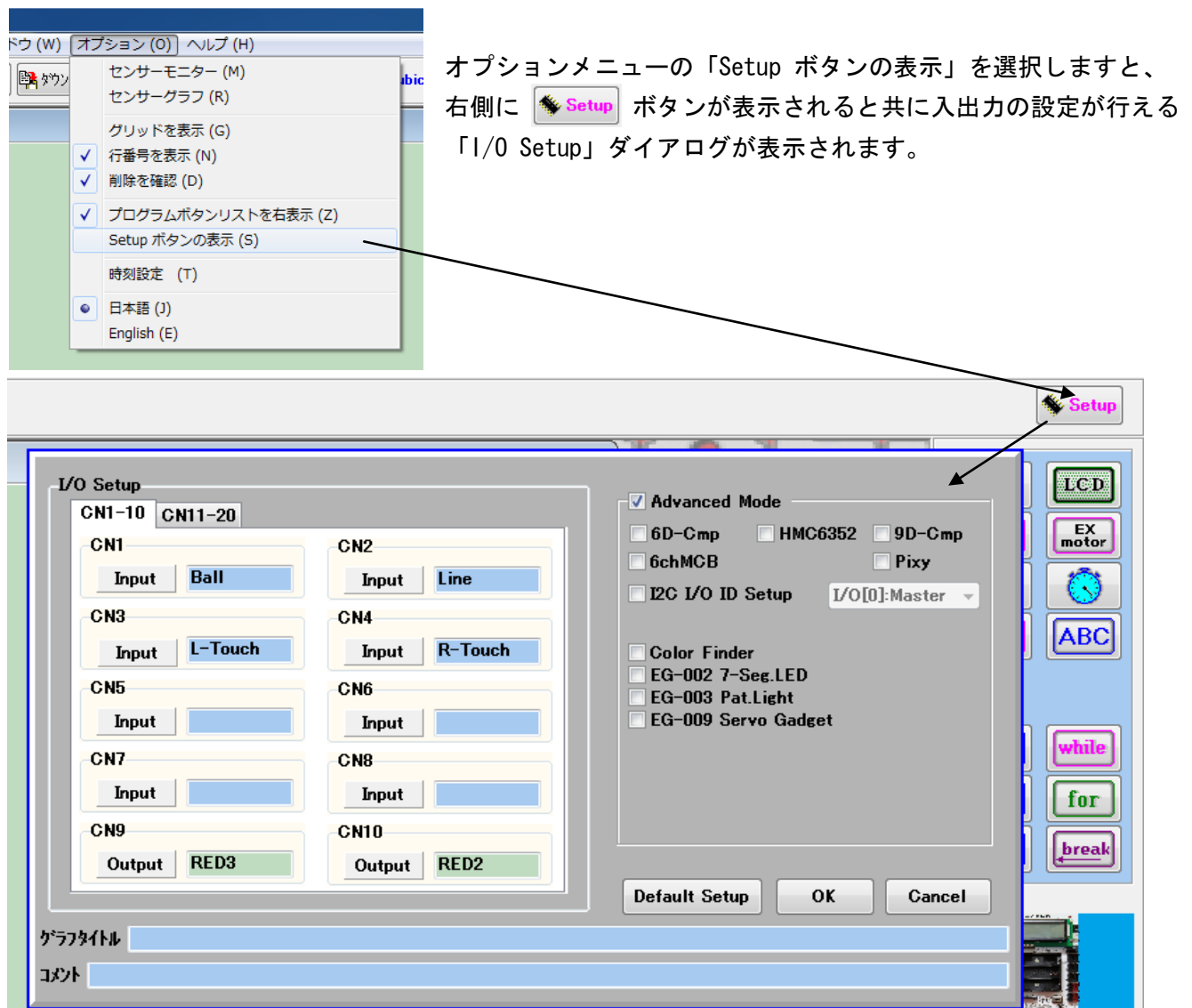
また CN9 と CN10 はデジタル出力で、ロボット本体の RED3、RED2 の LED に接続されています。

これらの端子は設定を変えることで、Output 設定にもできます。

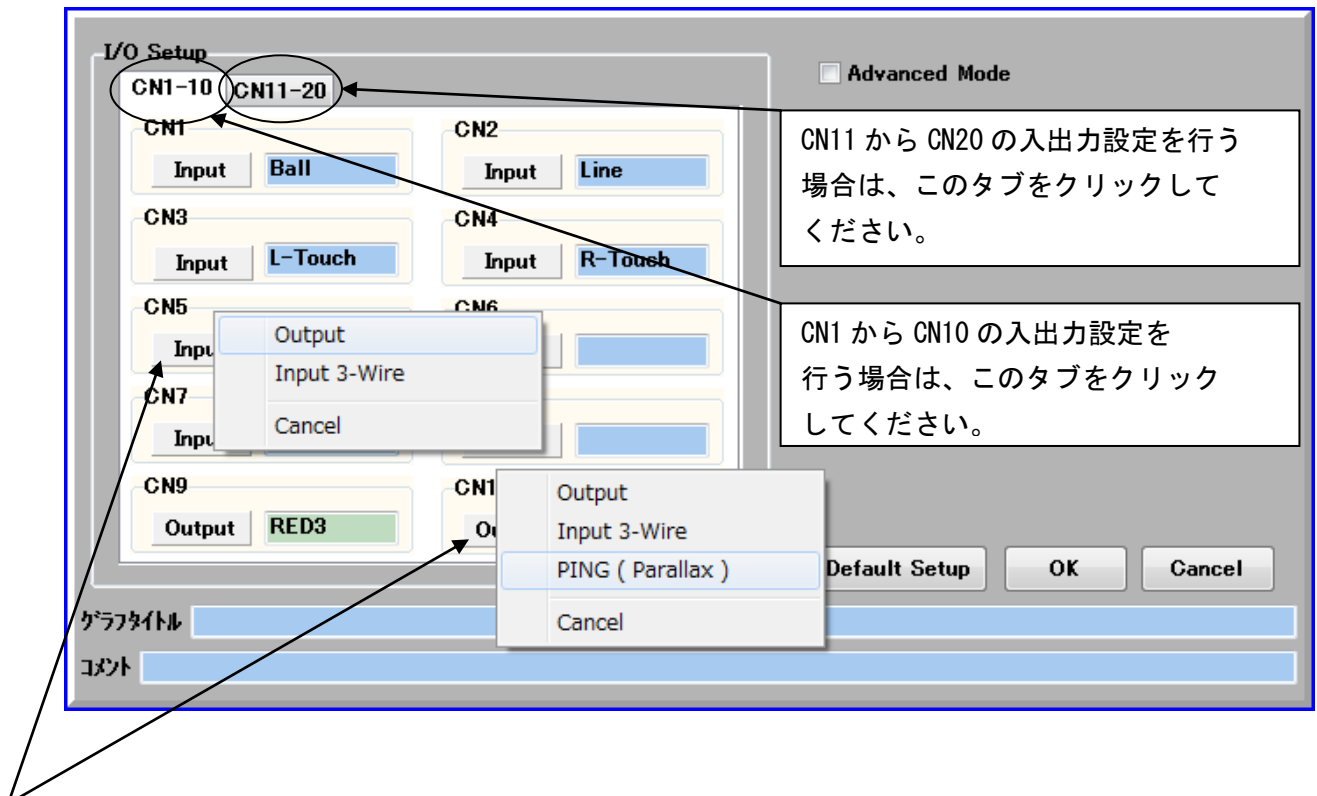
また、CN7 と CN8 はサーボモーター制御、CN7～CN10 は超音波センサーを接続することもできます。

その他にも 4ch・6ch モーターコントロールボードの制御ボタン設定や、多機能電子コンパス (6D/9D-Compass/HMC6352)、複数台の C-cubic を I2C 接続できる機能など拡張機能の設定が行えます。

3.1 Setup ボタンの表示




3.2 入出力設定



「 Input 」 / 「 Output 」 ボタンをクリックすると入出力設定用のポップアップメニューが表示されます。

CN7とCN8はサーボモーター制御の設定もできます。

またCN7からCN10は超音波距離センサー (Parallax 製の PING) を接続できる機能もあります。
(この設定は、CN10からCN7へと逆に設定していく必要があります。)

Output 設定をすると、プログラムボタンリストにボタン  が追加表示されます。

Input 3-Wire は、標準のアナログ入力センサーです。

[Default Setup] ボタンは入出力設定を初期の状態に戻す場合に操作します。

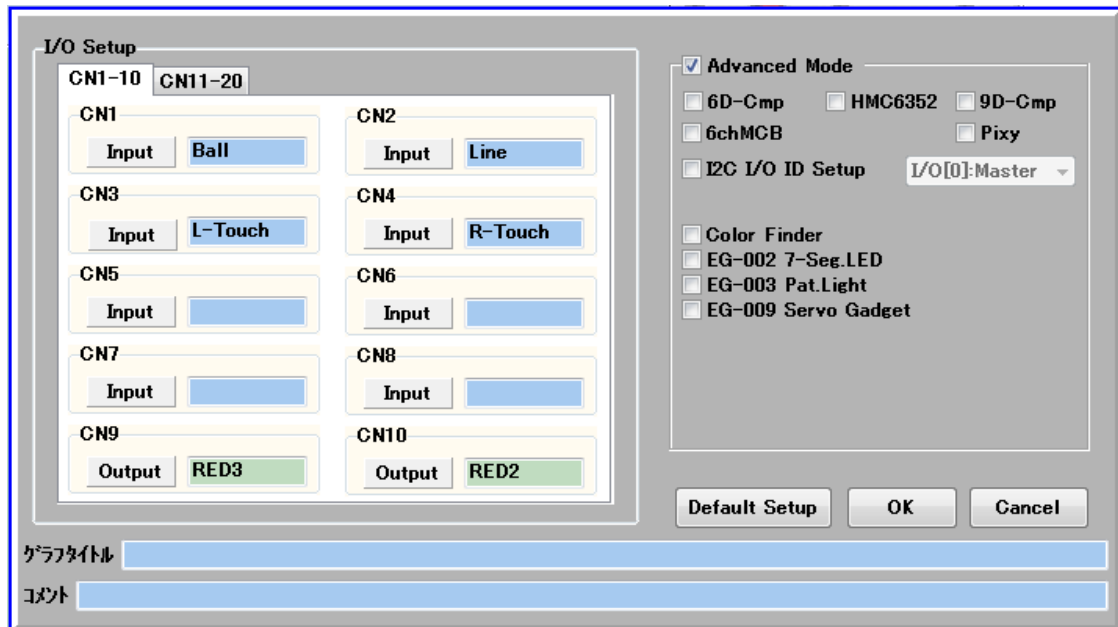
入出力の設定を変更した場合は、C-Style プログラムをビルドして C-cubic にダウンロードすることで、はじめて有効となります。

3.3 拡張機能の設定

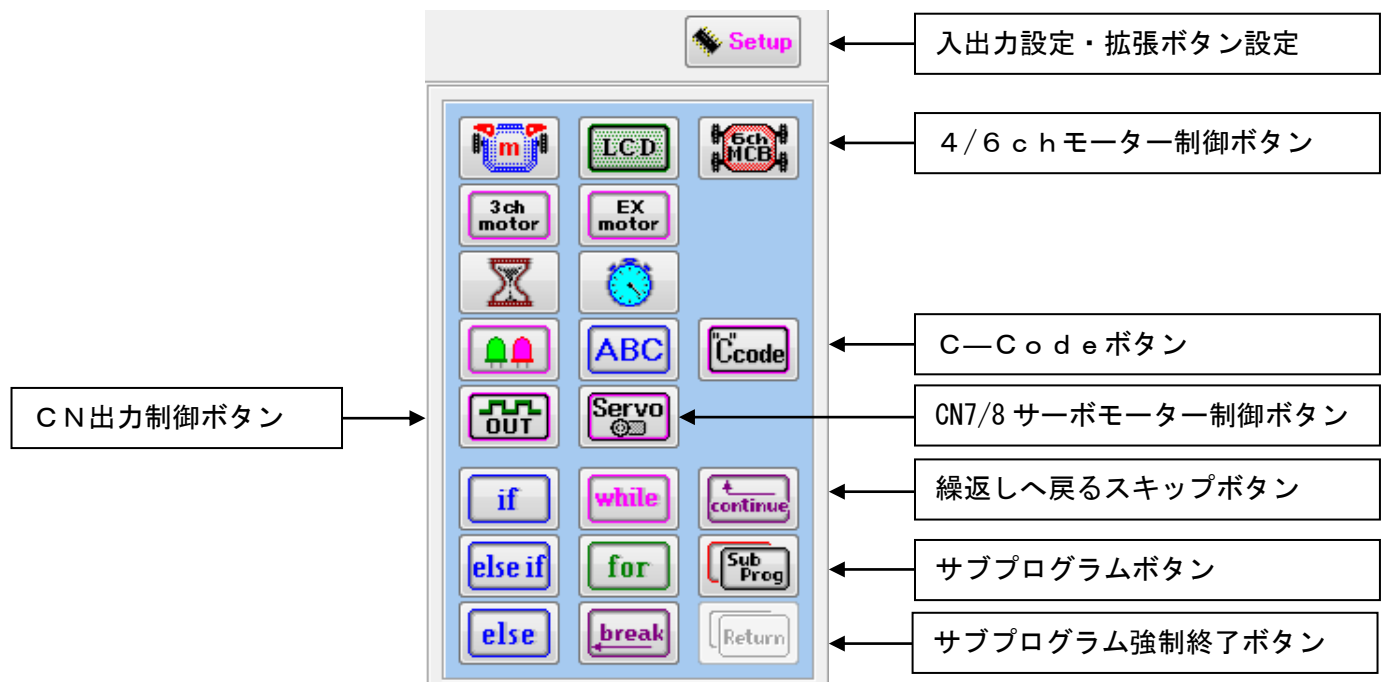
入出力設定ダイアログ内の右側にある ☐Advanced Mode にチェックを付けると拡張機能を設定するチェックボックスが表示されます。

6D/9D-Compass(多機能コンパス : DSR1603)、HMC6352(単機能電子コンパス : DSR1302)、4ch/6ch モーターコントロールボード設定、複数台の C-cubic を I2C 接続設定でセンサーボードとして使用する機能設定、EG シリーズ接続設定などの機能拡張の設定が行えます。

プログラムサイズは 246KB 使用できます。



拡張機能を設定するとプログラムボタンリストも下記のように変わります。

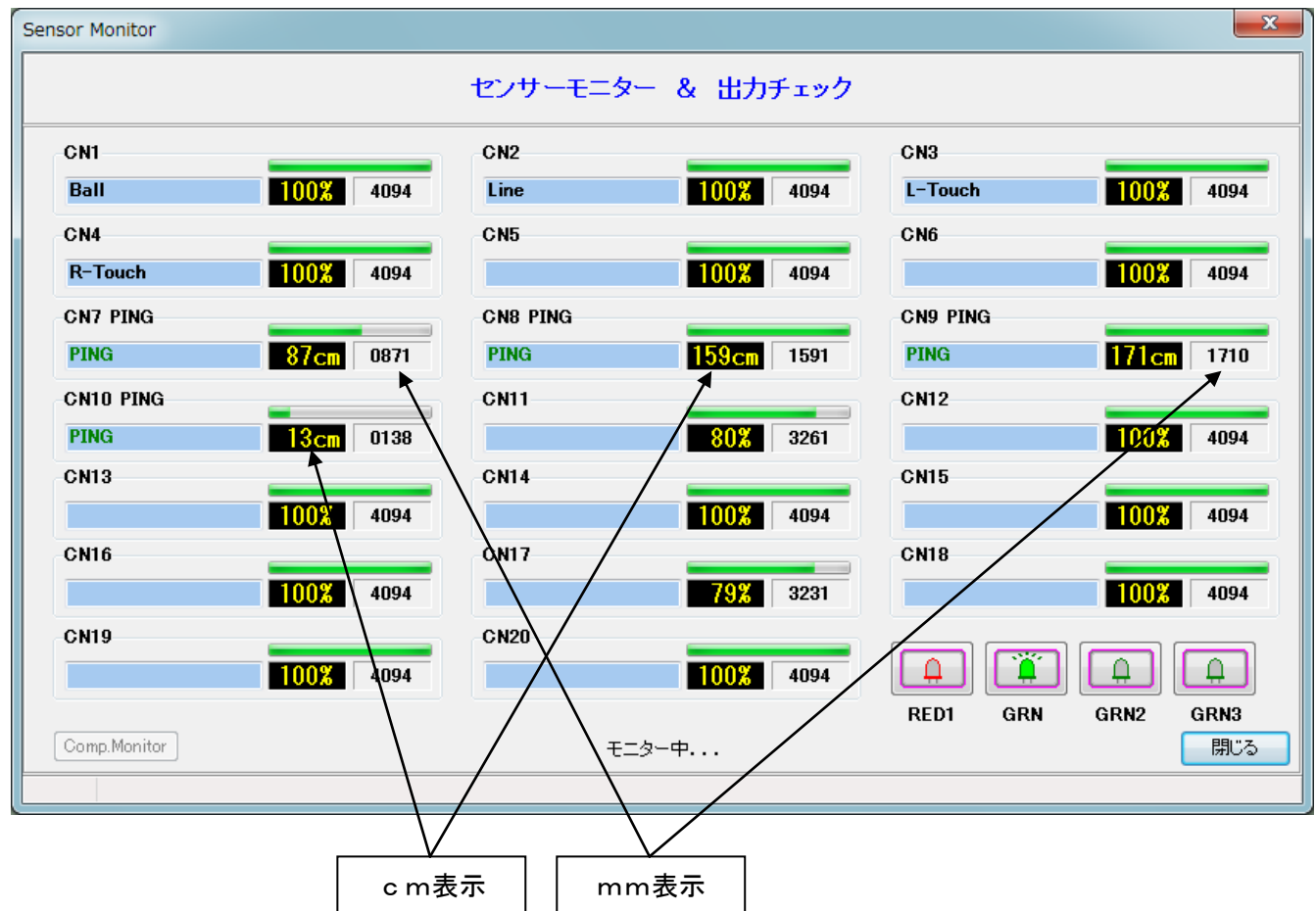


3.4 超音波距離センサーを使う

Parallax の PING 超音波距離センサーを接続する場合は、

入出力設定で、CN10～CN7 の Output/Input ボタンをクリックして PING に設定し、該当する CN 番号の PING センサーのチェックプログラムを作成してビルド、ダウンロードすることで、センサーモニターの値は % 表示から cm 表示に変わります。（約 3 cm から 3 m まで計測可能です）

CN10、CN9、CN8、CN7 を PING 接続に設定したモニター画面



CN10に接続したPINGが30cm以下で左回転するプログラム例



通常のセンサー入力の判定プログラムと同様に編集できます。
 上記の例の様に直接判定する場合は、cmの単位となります。
 $CN10 < 30\text{ cm}$

変数と比較する場合は、mmの単位で変数へ代入します。

変数A = 300 (300mm)

$CN10 < \text{変数A}$ ($CN10 < 30\text{ cm}$ と同じ内容となります)

変数への代入の場合も中身はmmの単位となります。

変数A = CN10

変数A < 300 (300mm : 30cm)

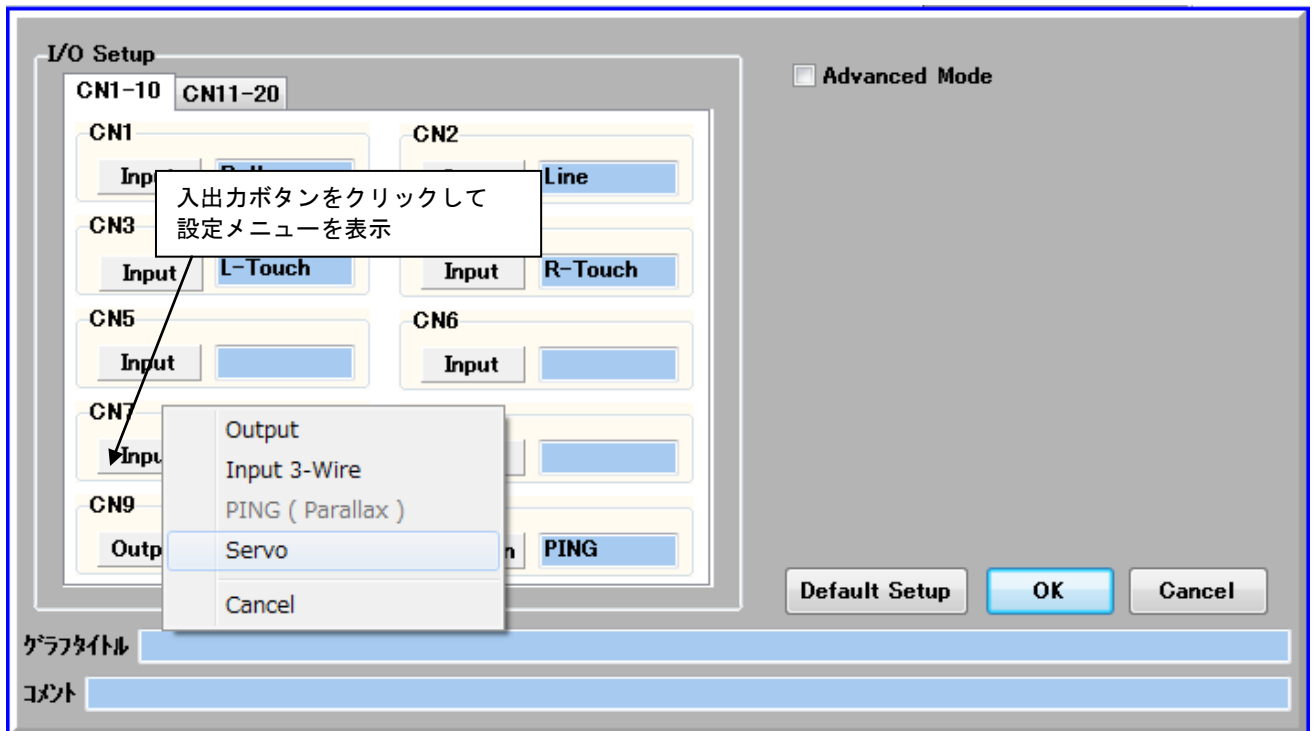
インストールしたC-Style for C-cubicのUserフォルダにPINGのサンプルプログラムがありますので参考になさってください。

Parallaxの超音波距離センサーPING (CN10~CN7に接続可能)

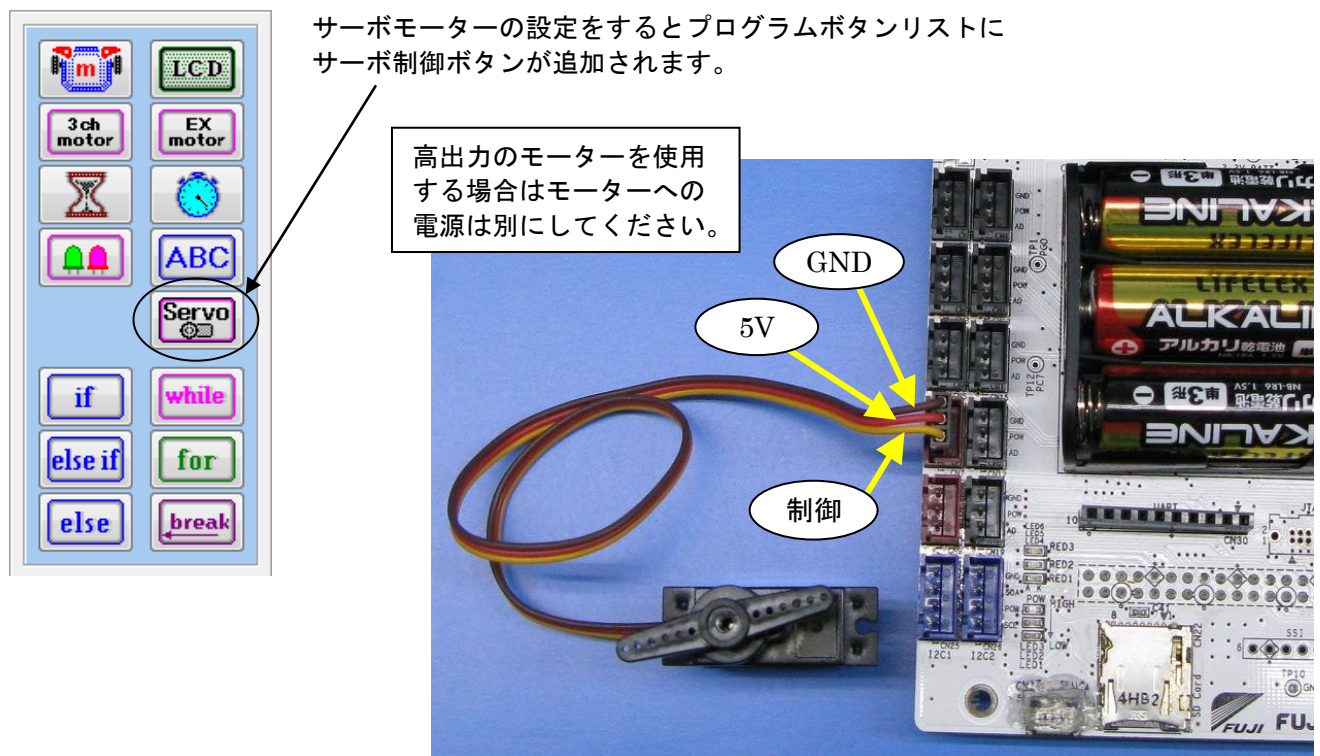


3.5 サーボモーターを使う

入出力設定で、CN7 と CN8 にはサーボモーター接続用の設定メニューがあります。

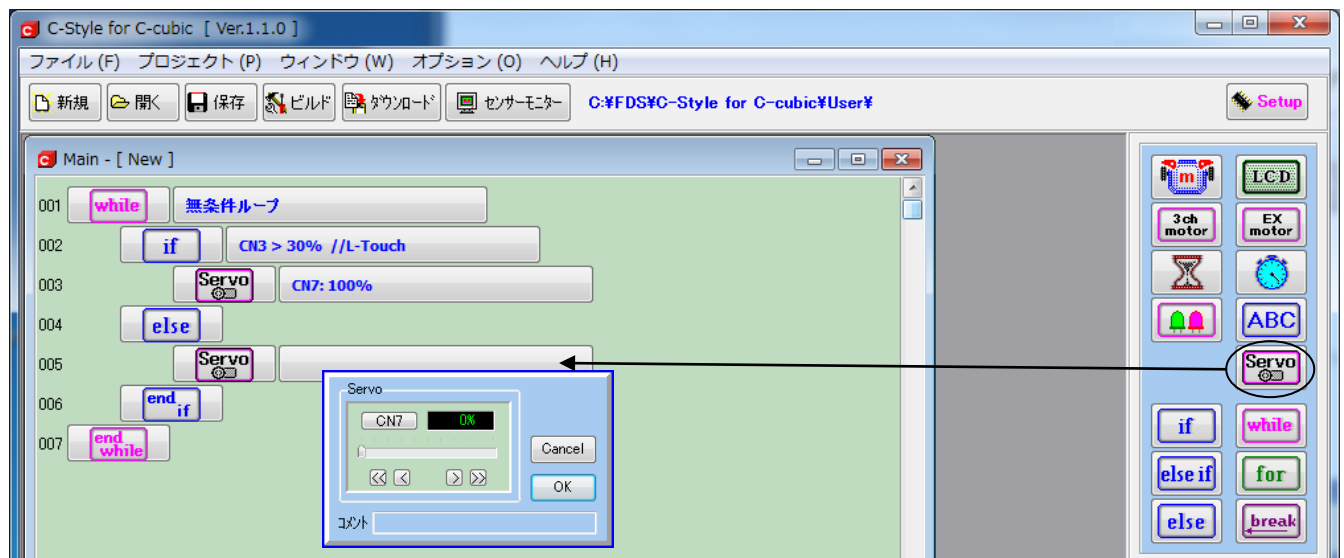


サーボモーターは 5V で動作するものが使えます。コネクターの 3 ピンの内 2 番目（中央）が電源端子になっていれば、ロボット本体の CN7 (CN8) に直接差し込むだけです。1 番ピンと 3 番ピンは制御信号と GND ですので、ロボット側のコネクターに一致するように接続します。



サーボモーター制御プログラムは、 ボタンを使ってプログラムできます。

サーボモーター制御プログラム例



この例では、左側のタッチセンサーを使ってサーボモーターを動作させています。

サーボ制御は 0% (500 μ 秒) から 100% (2500 μ 秒) で設定できます。

中央の位置が 50% (1500 μ 秒) です。

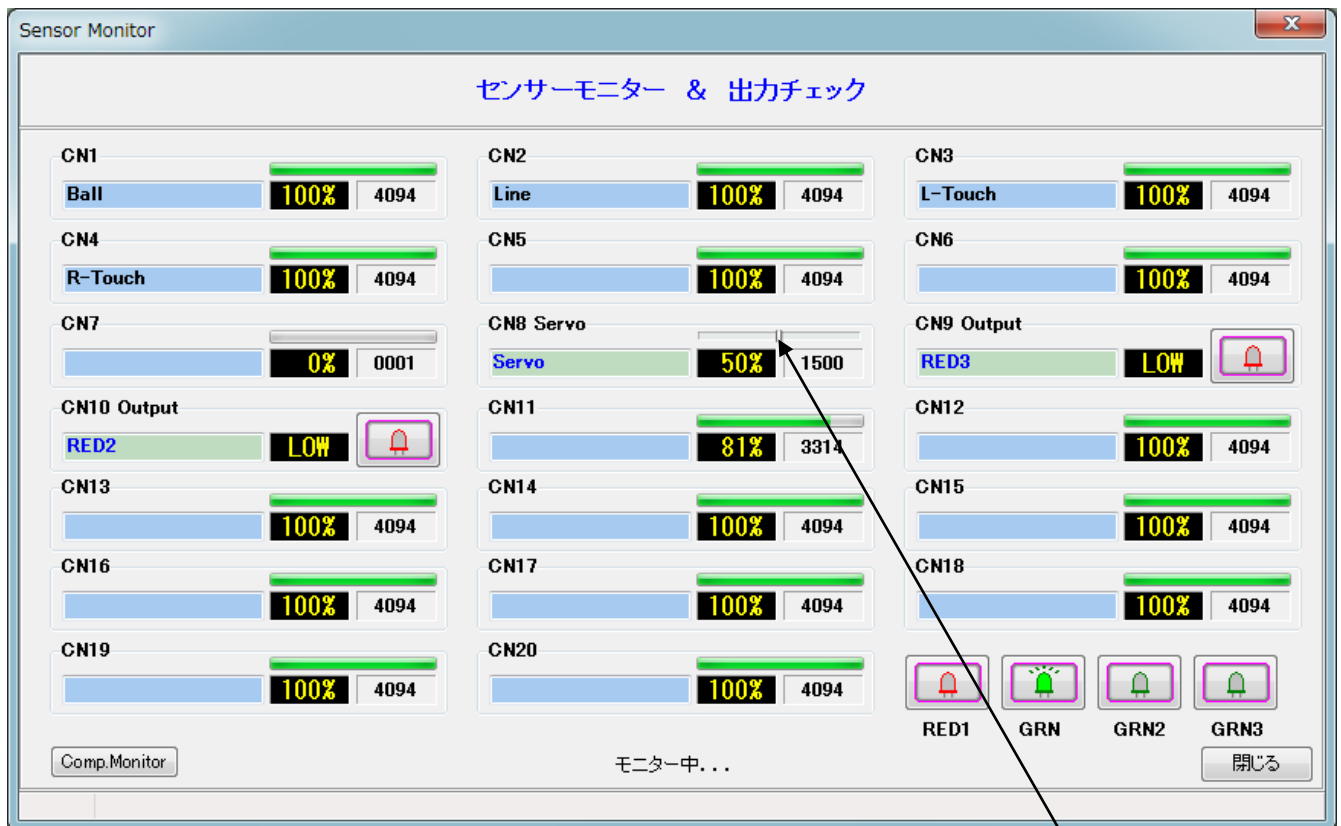
使用するサーボモーターによっては、最小値、最大値が上記の例とは異なる場合があります。

限界を超えた値を与え続けると故障の原因となりますので、センサーモニターで調べてください。

CN7 と CN8 の両方を使用する設定にしている場合は、設定ダイアログ内の「CN7」のボタンをクリックすると「CN8」の表示に変わり制御するサーボを選択することができます。

ロボットからの電源でサーボモーターを動かすと、ロボットの電源回路が破損する恐れがありますので、高出力のサーボモーターを使用する際は、必ず別電源でサーボモーターへ電源供給してください。

センサーモニター画面

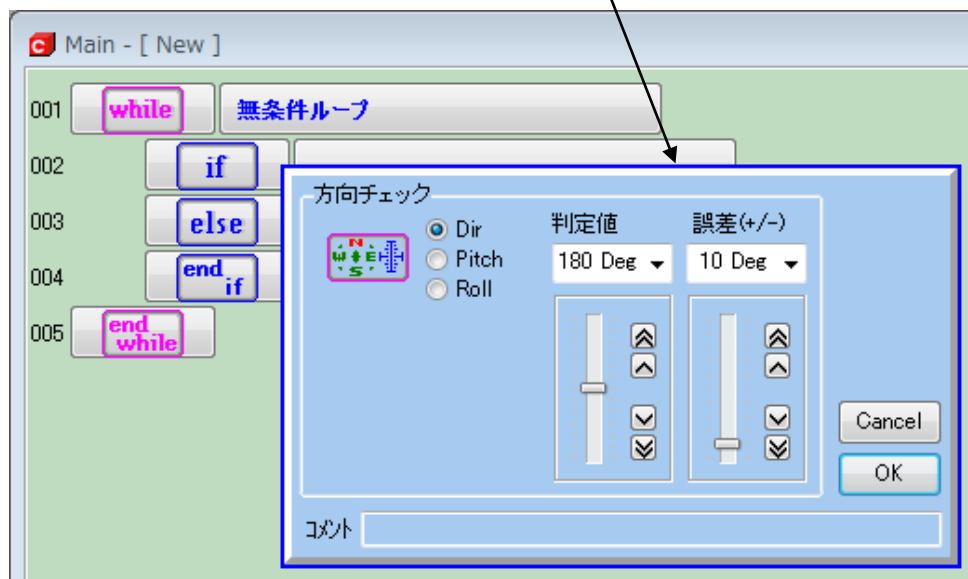


スライダーに連動してサーボモーターが動作します。

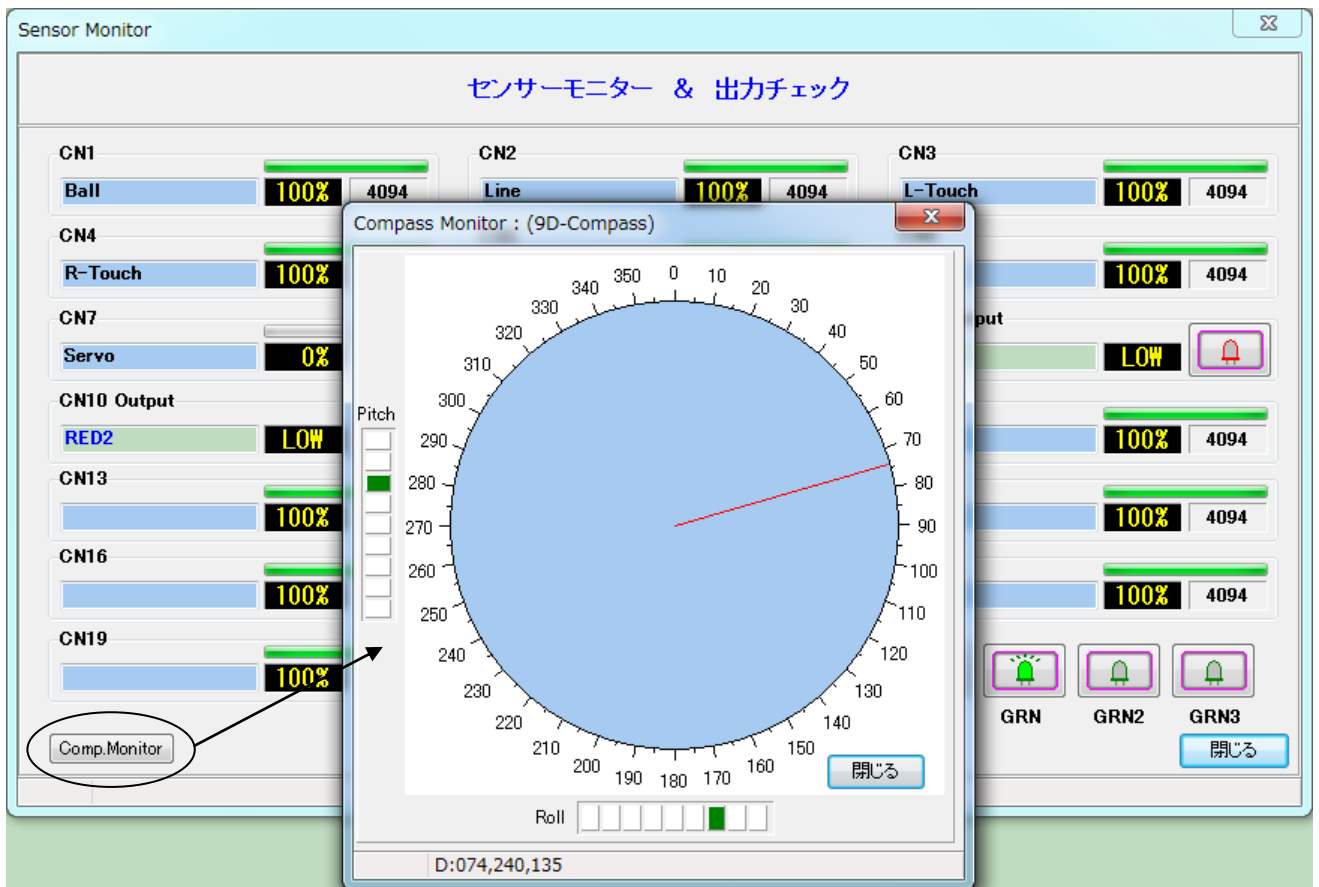
サーボモーター制御機能を設定したプログラムをダウンロードすれば、センサーモニターでの動作確認が行えます。

3.6 6D/9D-Compass（多機能電子コンパス：DSR1005/DSR1603）を使う

多機能電子コンパスを使用する場合は、6D-Cmp または 9D-Cmp にチェックを付けます。



設定画面の 6D/9D-Cmp または HMC6352 にチェックの付いたプログラムをダウンロードすると、センサーモニター画面の左下の「Comp. Monitor」ボタンが有効となります。

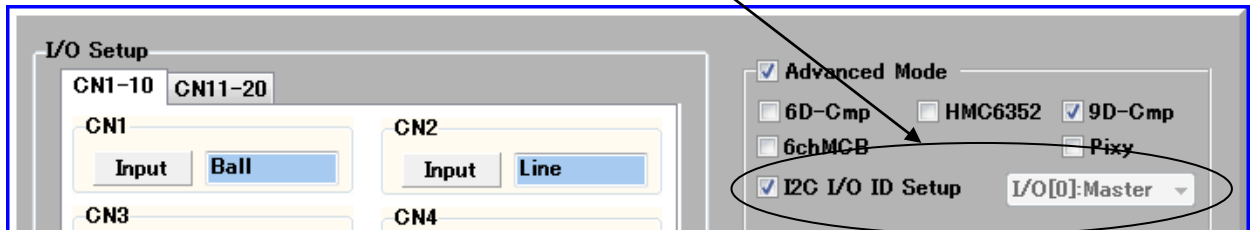


センサーモニターの「Comp. Monitor」ボタンをクリックすると現在の方向を表示するダイアログを表示することができます。



3.7 複数台のガジェット を接続する

入出力設定画面の I2C I/O ID Setup にチェックを付けると最大 8 台までのガジェットを接続して I/O センサーボードとして使用できるようになります。

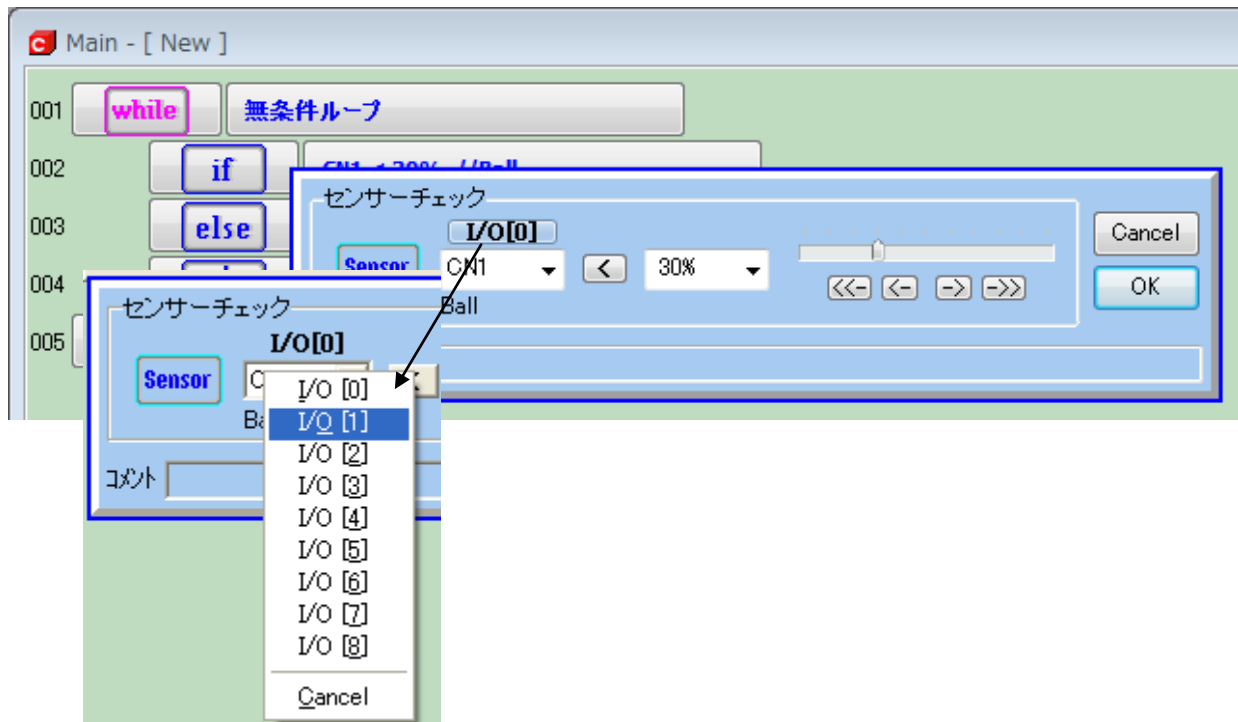


この機能は、通常 C-cubic は 1 台当たり最大 20 個のセンサー (CN1～CN20) を使うことができますが、I2C という通信仕様のコネクタ (4pin) を使って別の C-cubic と接続してセンサーを増やすことができます。

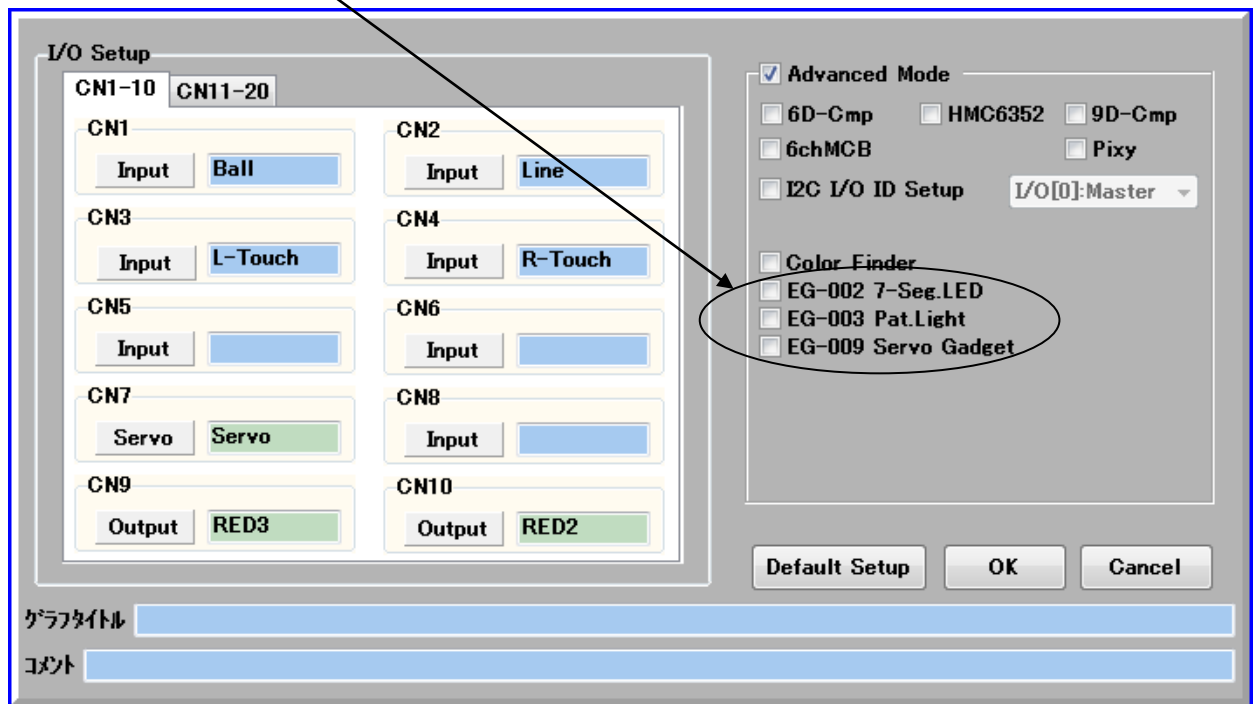
センサー情報を一括管理する親の C-cubic のことを Master (I/O[0]) と呼び、センサー情報を親へ引き渡すガジェットのことを Sub I/O (I/O[1]～I/O[8]) と呼びます。

Setup 画面では、Master の設定か、Sub I/O の何番の ID : 1～8 に設定するかを決めてからプログラム編集を行います。

通常、Sub I/O のガジェットのプログラムは、この Setup 画面で設定した ID (1～8) だけでビルドしてダウンロードすれば、電源を入れるだけ (スタートボタンは押さなくても良い) で、Master 側のプログラムによって Sub I/O のセンサー情報を取出して条件判定する事ができます。



3.8 EG シリーズを使う



C-cubic には、EG シリーズと称して拡張パーツが使える機能が実装されています。

現在は、3 機種のパーツが用意されています。

EG-002 : 4 桁の 7Seg.LED 表示器

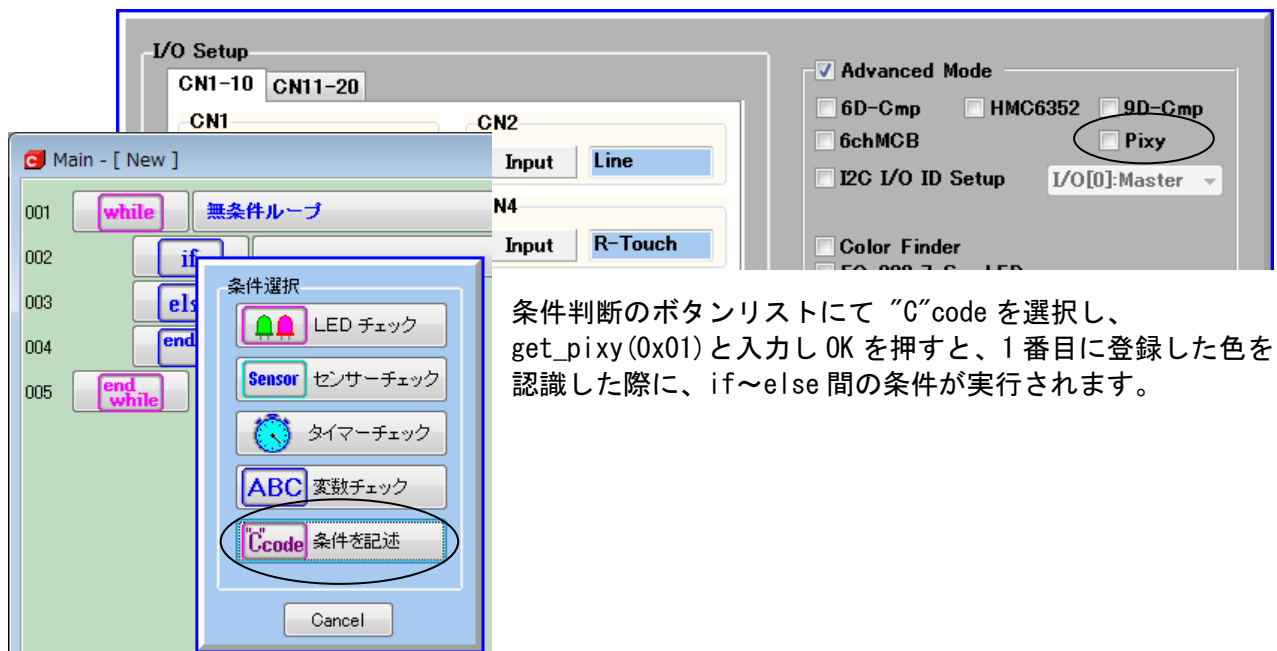
EG-003 : 赤、橙、緑色の 3 色 LED ミニパトライト

EG-009 : 1 軸サーボモータコントローラ

これらの制御コードは、C-Code で記述する必要があります。

3.9 Pixy カメラを使う


Pixy カメラを使用する場合は、Pixy にチェックを付けます。

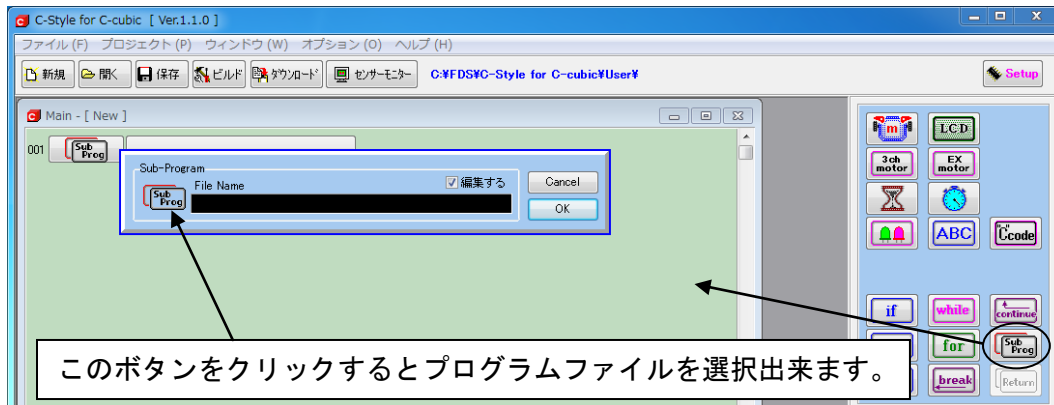


条件判断のボタンリストにて “C”code を選択し、
get_pixy(0x01) と入力し OK を押すと、1 番目に登録した色を
認識した際に、if~else 間の条件が実行されます。

4 サブプログラムの説明

4.1 サブプログラムボタンを置く

I/O Setup（入出力設定）ダイアログ内の□Advanced Mode にチェックを付けて「OK」ボタンで Setup 画面を閉じると、プログラムボタンリストが3列表示となり  ボタンが使用できるようになります。




同じ処理を何回も再利用したい場合や、プログラムを見やすくする為に、一定の処理プログラムを一つのプログラムボタンにまとめることができます。（サブルーチンと言います）

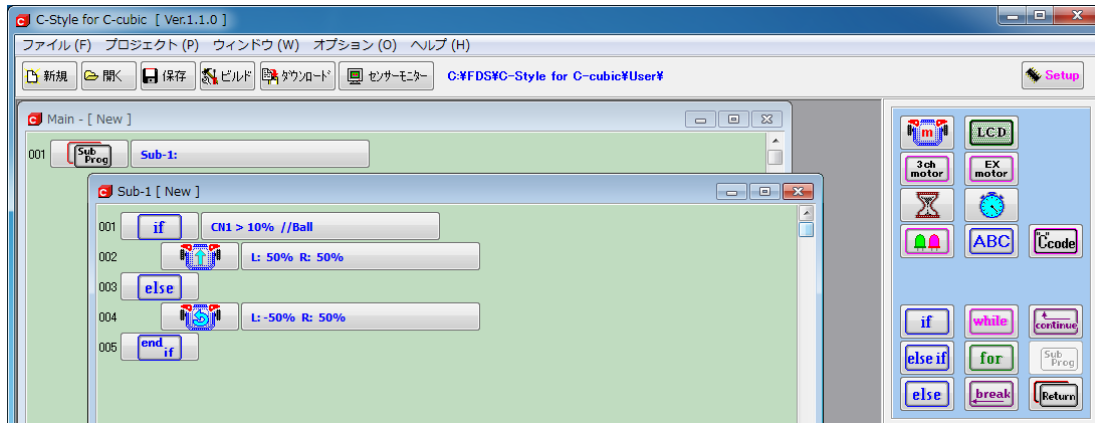
サブプログラムは30種類作成することができます。

但し、サブプログラムの中にサブプログラムボタンを置くことはできません。

新規にサブプログラムを作成する場合はボタンを置いてから「OK」をクリックします。

既に作成されたサブプログラムを指定したい場合は、Sub-Program ダイアログ内の  ボタンをクリックするとファイルから選択できます。

4.2 サブプログラムの編集



サブプログラムの編集フォームが表示されたら、メインフォーム同様にプログラムボタンリストからプログラムボタンを選択して、サブプログラムの編集を行います。

編集されたサブプログラムを保存するには、サブプログラムフォームをアクティブな状態にしてから、「保存」ボタンをクリックします。

アクティブな状態にするには、保存したいフォームのタイトルバーをマウスでクリックすると、他のフォームより最前面に表示されます。

サブプログラムはメインプログラムが保存されている同じフォルダ内に保存してください。

保存したファイル名が、メインフォームのサブプログラムボタンのコメント表示となります。

Setup 画面で設定するオプションは全てのメイン・サブプログラムで同じ設定にしてください。

メモ

メモ

お問合せ



〒660-0892 兵庫県尼崎市東難波町 5 丁目 21 番 14 号

TEL 06-6401-1211 (FAX 06-6401-1244)

URL <http://www.fdsnet.co.jp/>

Email Ccubic@fdsnet.co.jp

※C-Style は株式会社ダイセン電子工業が作成しています。