ロボットプログラミングアプリ

C-Style 操作編

株式会社ダイセン電子工業
DAISEN

REV250623

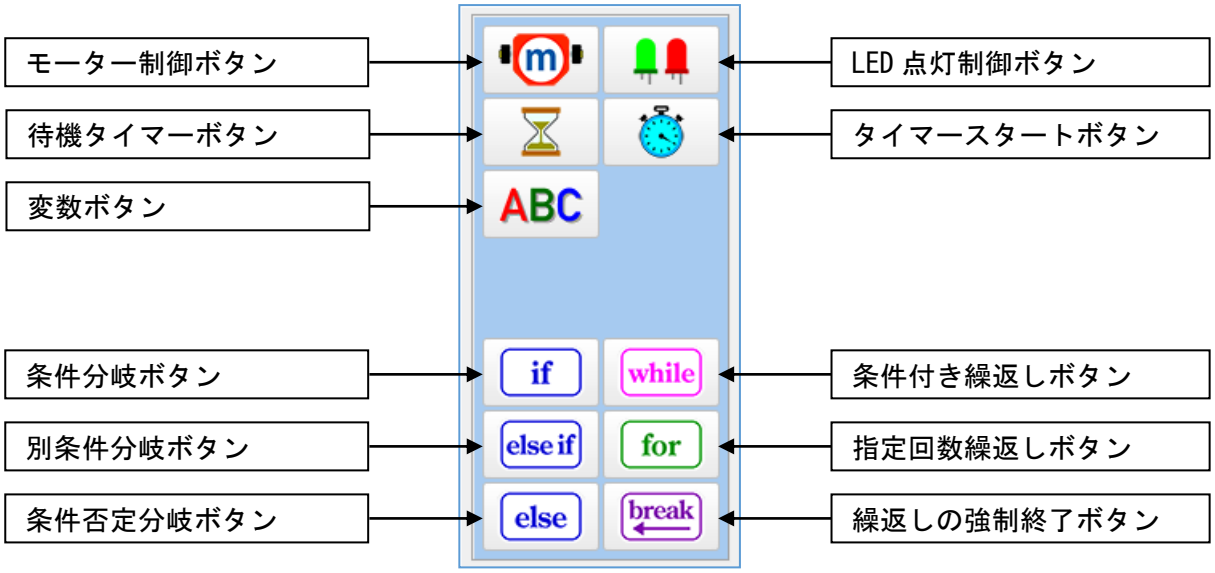
目　次

C-Style 操作編（本書）

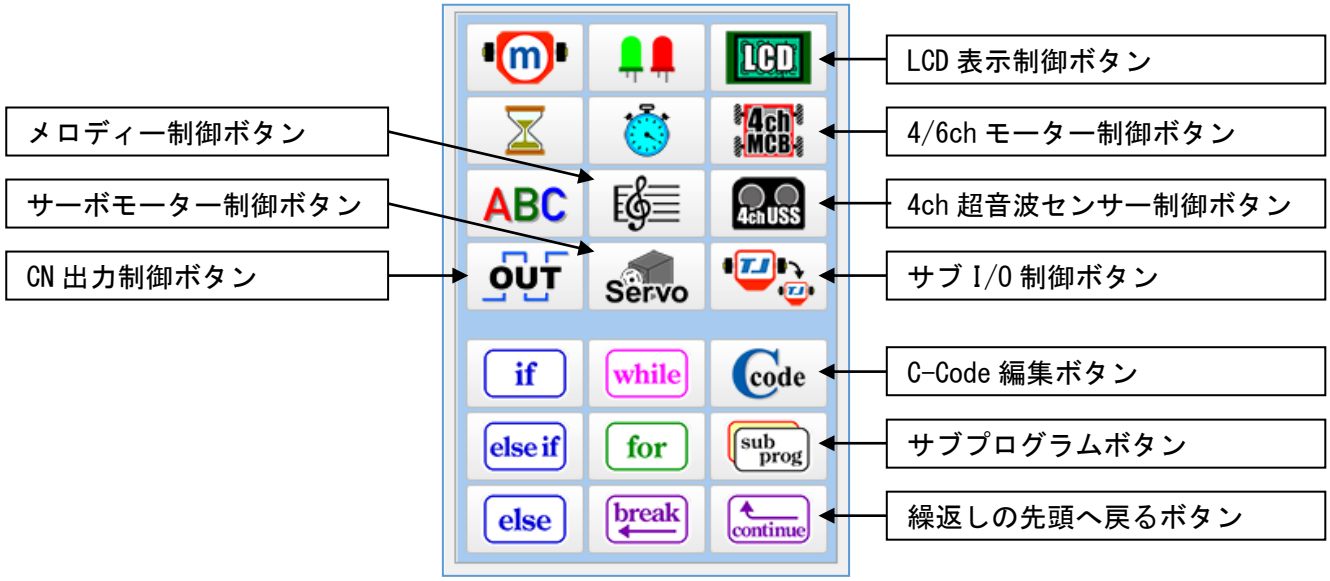
１．　プログラムボタンの説明

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



機能拡張時のプログラムボタンリスト

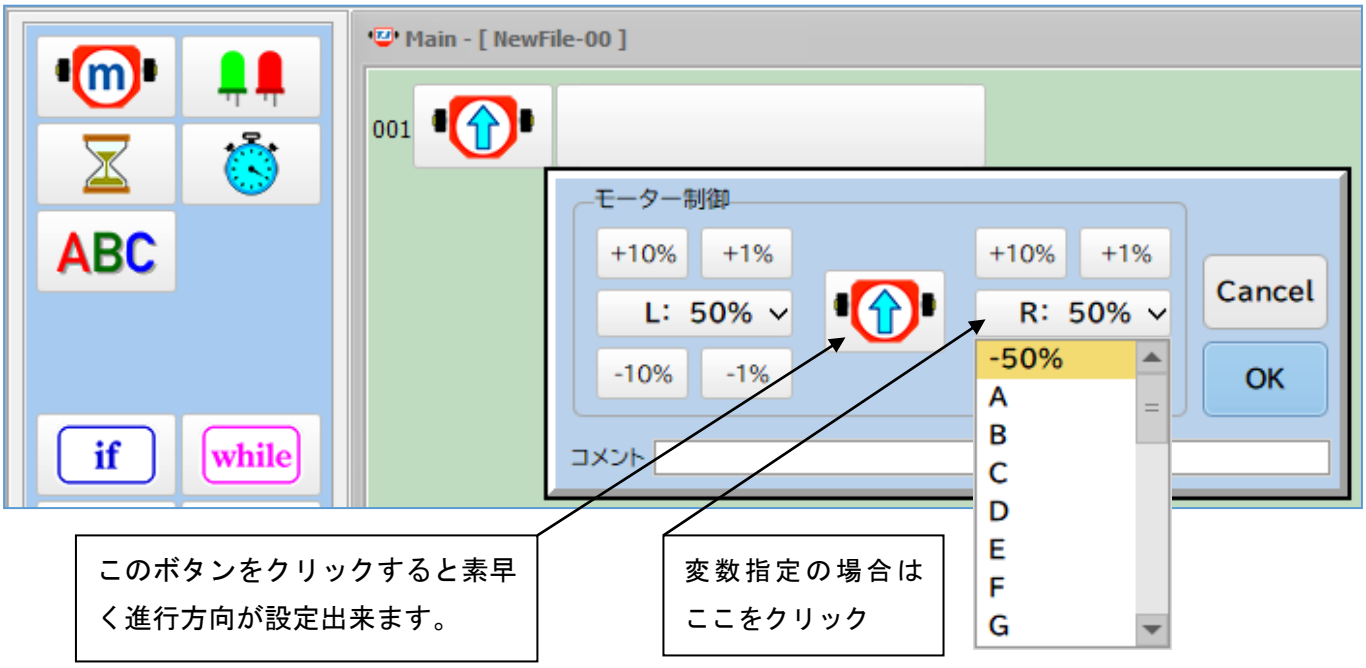


拡張機能については、「３. 入出力設定と拡張機能の設定」のページを参照して下さい。

C-Style プログラムのボタン名称

1-2. モーター制御ボタン 

ロボットのモーターへの速度や回転方向を決めます。(L:左側、R:右側)



[+10%], [+1%]ボタンをクリックすると　1～ 100%の前転速度となります。
[-10%], [-1%]ボタンをクリックすると -1～-100%の後転速度となります。
0%はモーターが停止します。
左右の速度を+/-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が簡単に設定することが出来ます。左右どちらかの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、反対側のモーター速度も同じ値に設定されます。

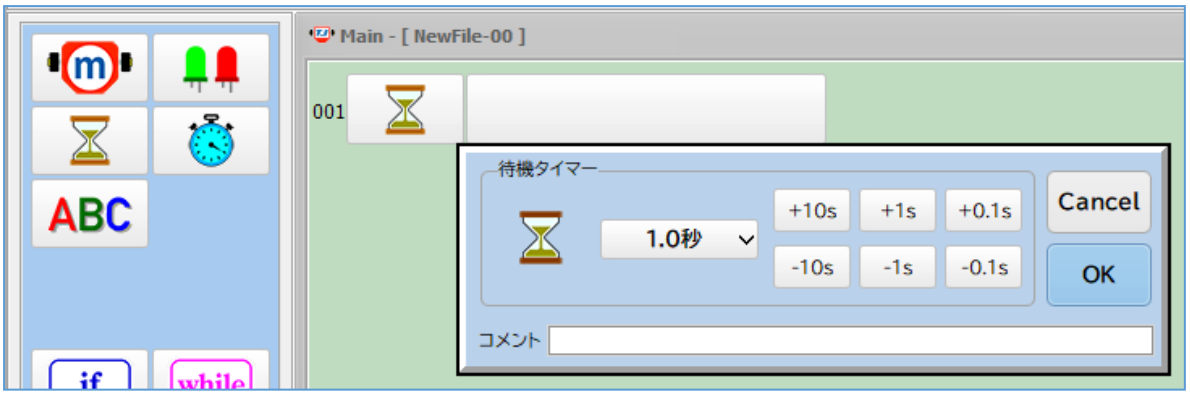
左右の速度が同一方向で 20%以上の場合はモーターアイコンが旋回アイコンの表示となります。

速度指定が変数の場合は、方向を示すアイコン表示にはなりません。

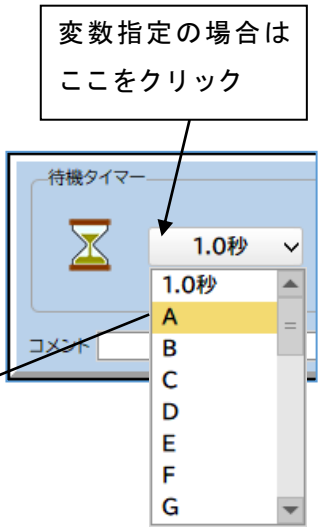
※変数の操作方法は「1-6. 変数ボタン」を参照

1-3. 待機タイマーボタン 

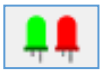
プログラムの待ち時間を設定します。



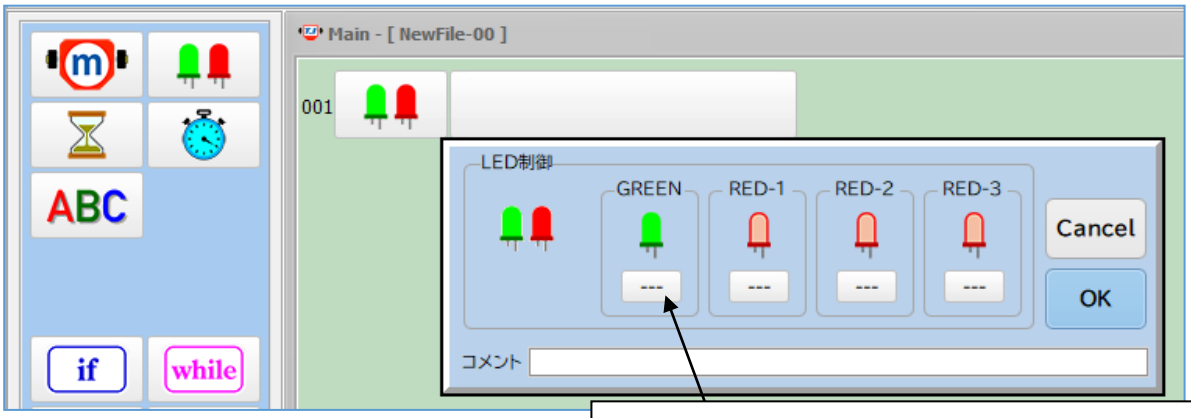
0.1 秒から 60.0 秒までの待ち時間を設定します。
待ち時間を変数で設定することも出来ます。
変数へ待ち時間を設定する場合は、1000 倍したミリ秒の単位で設定します。
例えば、1.5 秒の待ち時間を変数へ設定する場合は、A=1500 として
待ち時間表示をクリックして「変数 A」選択します。

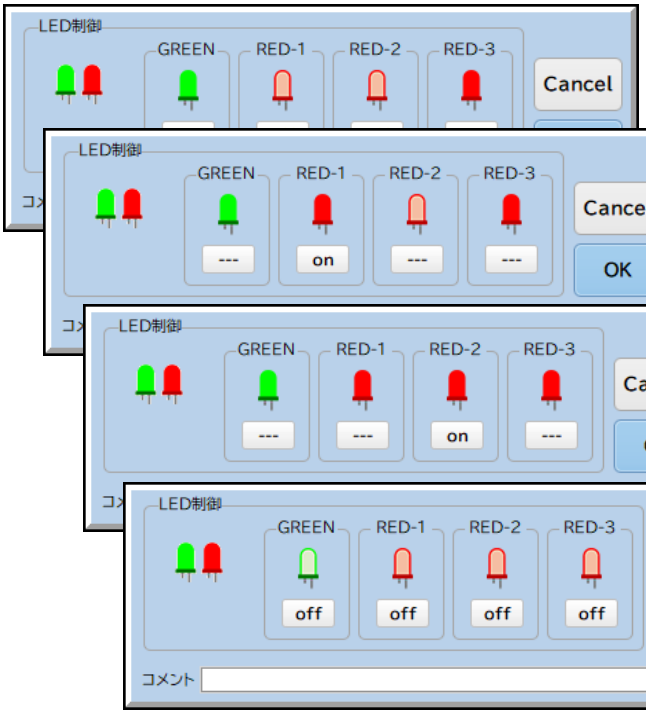


待機タイマーボタンは、簡単に待ち時間をプログラムすることができますが、待っている間は、他のプログラムボタンを置いて制御することが出来ません。
待っている間も他のプログラム制御したい場合は、タイマースタートボタンとタイムチェックを組み合わせて行います。(1-10. タイマーチェックを参照)
※変数の操作方法は「1-6. 変数ボタン」を参照

1-4. LED 制御ボタン 

緑 LED と赤 LED1～3 の LED4 個を同時に制御します。





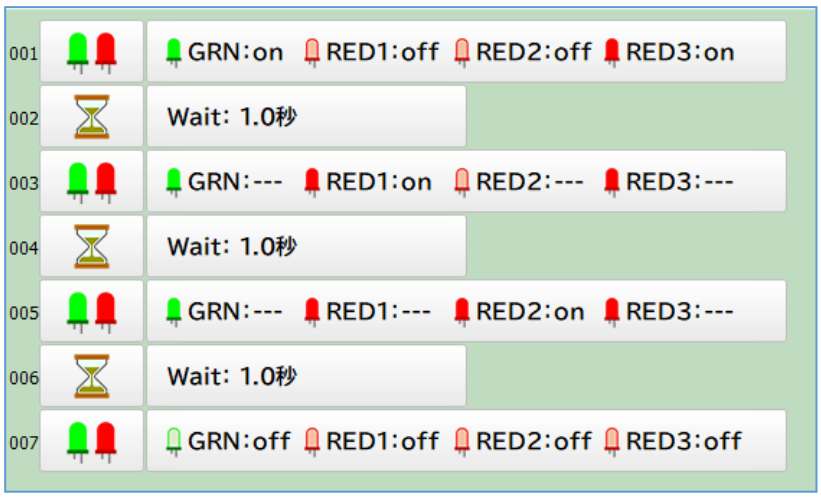
GREEN と RED3 が点灯で、他は消灯

RED1 だけ点灯で、他は現状維持

RED2 だけ点灯で、他は現状維持

全ての LED が消灯

LED 点灯推移のプログラム例

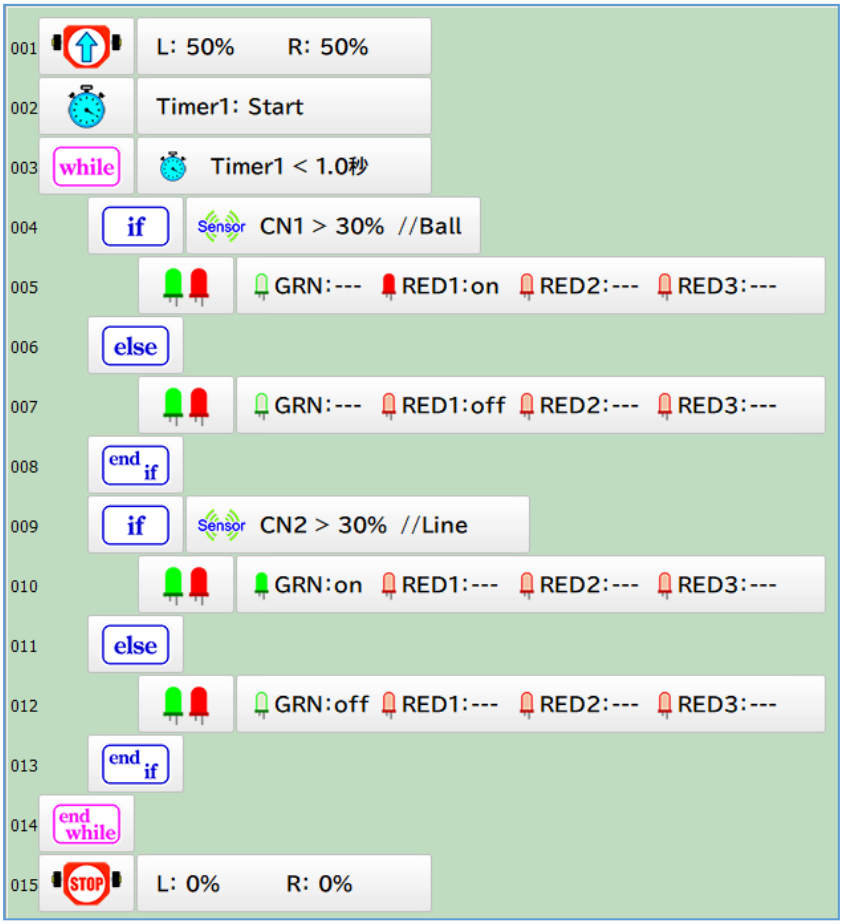


1-5. タイマースタートボタン 

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。

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、停止する機能はありませんので、再計測する場合は再スタートさせます。

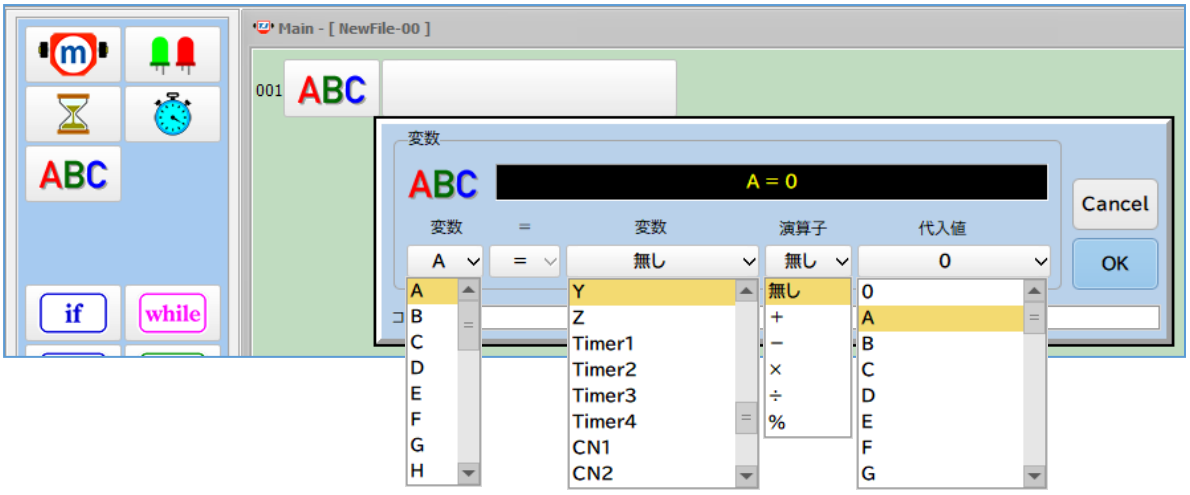
タイマーを使ったプログラム例



この例は、導入編「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の待機タイマーの代わりにタイマーを使って同じ動作を実現しています。
待機タイマーを使用した場合ではただ1.0秒間待つだけで、他のプログラムを実行することが出来ませんでした。
タイマーを使った場合では1.0秒間に他のプログラムを実行することが出来ます。

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30%以上の判定でRED1のLEDを点灯し、CN2のラインセンサーが30%以上の判定で緑LEDの点灯することが出来ます。

1-6. 変数ボタン ABC

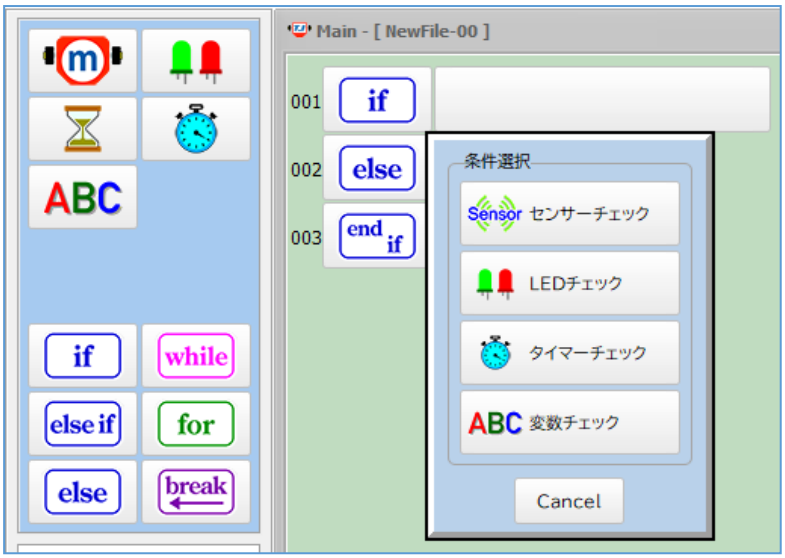


変数は A～Z の 26 個が使用出来ます。
演算子と＋、－、×、÷、余り計算などが行えます。
またタイマーの値や CN1 から CN10 に接続されたセンサー値も変数へ代入することが出来ます。

◇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
“T = Timer1”	変数 T に Timer1 の値を代入（変数 A は 0～65535 の値になる）
“C = CN1”	変数 C に CN1 のセンサー値を代入（CN1 は 0～1023 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 は以前の値の半部になる）
“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 1 となる）
“A = B - CN6”	変数 A に変数 B から CN6 のセンサー値を引いた値を代入

1-7. 条件分岐ボタン if else if



if　イフと呼びます。
判定する条件を選択します。

else if　エルスイフと呼びます。
判定する条件を選択します。

else　エルスと呼びます。

end if　エンドイフと呼びます。

「if」と「else」と「end if」の組合せで必ず配置されます。
「else」は必要に応じて削除出来ます。挿入の場合は「end if」の前に１個だけ置けます。
「else if」は「if」と「else」または「end if」の間に幾つでも置けます。
「if」と「else if」を置いた時に、センサーチェック（ボールセンサー、ラインセンサー、タッチセンサーなど)、LED の点灯状態をチェック、タイマーチェック、変数チェック等の判定を行う為の条件選択が表示されます。

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。
条件が不成立の場合は、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。

「if」を切取りまた削除する場合は「end if」までのブロック単位となります。

1-8. 条件付き繰返しボタン while

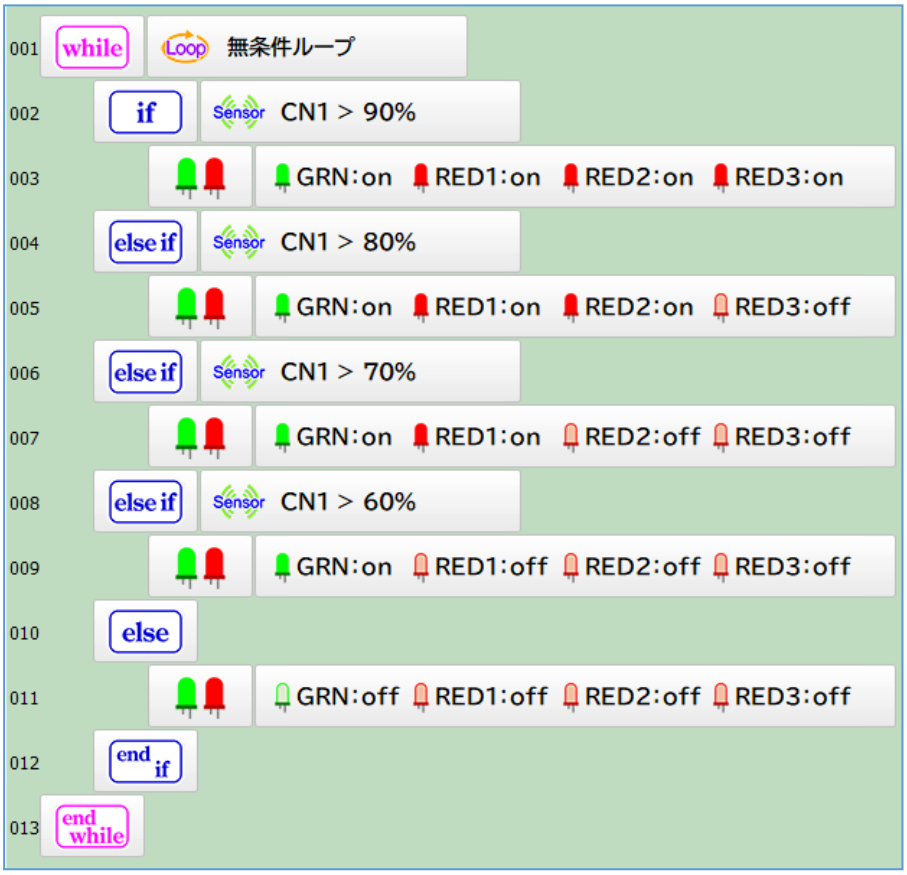
while ホワイルと呼びます。
繰返す条件を選択します。

end while エンド ホワイルと呼びます。
条件が成立している間、「while」と「end while」の間に置かれたプログラムボタンが繰返し実行されます。
「while」と「end while」ボタンの組合せで必ず配置されます。

break ブレイクと呼びます。
繰返しブロックの中にだけ置けるボタンで、このボタンに出会うとその位置から繰返しは強制終了され「end while」の次の行から実行されます。

continue コンティニューと呼びます。 繰返しブロックの中にだけ置けるボタンでこのボタンに出会うと、繰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)
このボタンは拡張機能の設定（Setup 画面の“Advanced Mode”にチェックを付ける）で表示されます。

while, if, elseif, else プログラム例



while～end while(001～013)の範囲を無条件で繰り返す
002～003：CN1 のボールセンサーが 90%以上で全ての LED を点灯

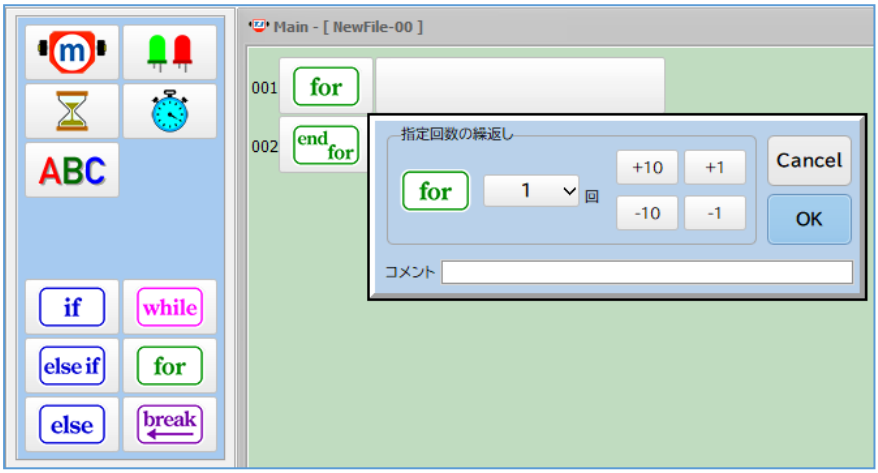
004～005：CN1 のボールセンサーが 80%以上で RED3 の LED を消灯

006～007：CN1 のボールセンサーが 70%以上で RED2, RED3 の LED を消灯

008～009：CN1 のボールセンサーが 60%以上で RED1, RED2, RED3 の LED を消灯

010～011：上記の条件以外の場合は全ての LED を消灯

1-9. 回数指定の繰返しボタン for



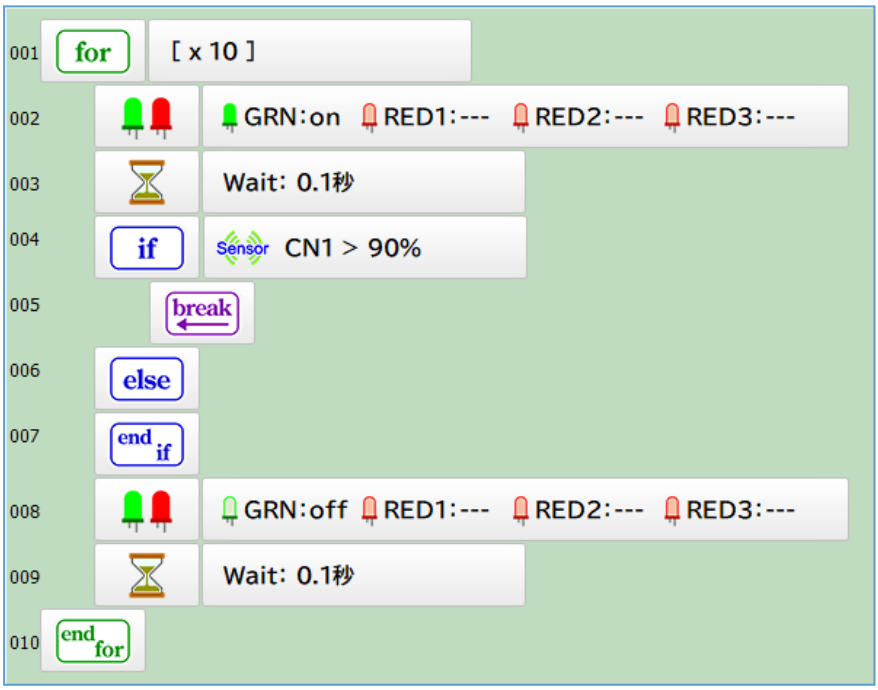
for フォアと呼びます。
繰返す回数を指定します。

end for エンド フォアと呼びます。
指定された回数、この間に置かれたプログラムボタンを実行します。

break ブレイクと呼びます。繰返しブロックの中にだけ置けるボタンで、このボタンに出会うと、繰返しは強制終了され、その位置から「end for」の次の行へ実行は移されます。 （「break」から「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。 繰返しブロックの中にだけ置けるボタンでこのボタンに出会うと、繰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）
このボタンは拡張機能の設定（Setup 画面の“Advanced Mode”にチェックを付ける）で表示されます。

for と break を使ったプログラム例

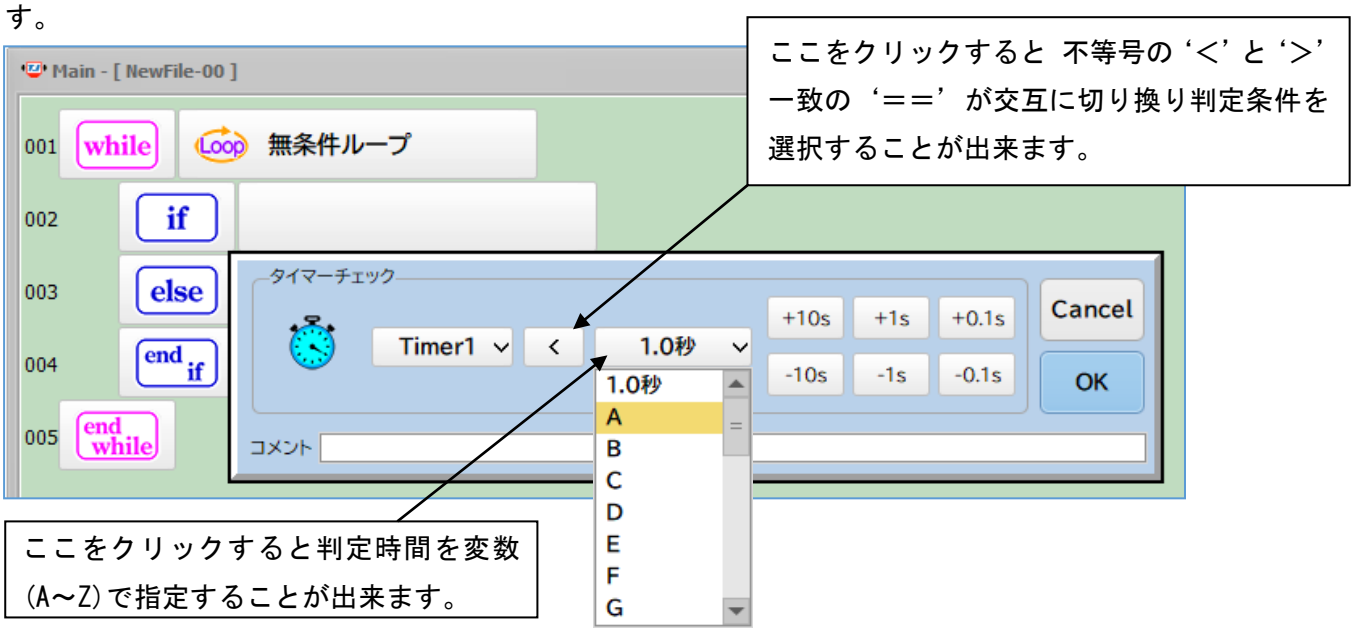


for～end for (001～010) までのプログラムを 10 回繰返します。

通常は GRN-LED が 0.1 秒間隔で点滅しますが、
004:CN1 のボールセンサーが 90%以上の場合 break して繰返しを中断しますので、この例では GRN-LED が点灯したままとなります。

1-10. タイマーチェック

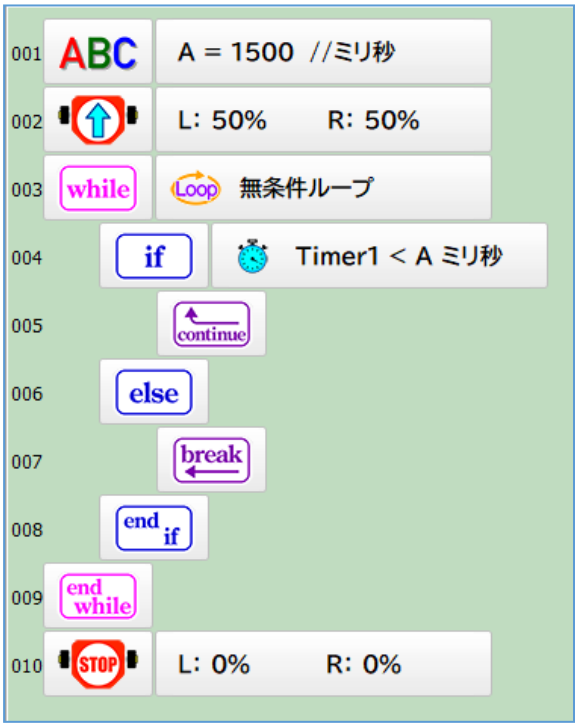
事前にタイマースタートしたTimer1～Timer4のいずれか選択してその経過時間をチェックして条件判定します。



判定例

“Timer1 ＜ 0.1秒より下　　（ 0.1秒は含みません ）
“Timer2 ＞ 1.5秒より上　　（ 1.5秒は含みません ）
“Timer1 ＞ 変数A より上”　　（ 変数Aに代入した値は含みません）
“Timer2 ＜ 変数A より下”　　（ 変数Aに代入した値は含みません）

変数を使ってロボットを1.5秒間前進して停止するプログラム例



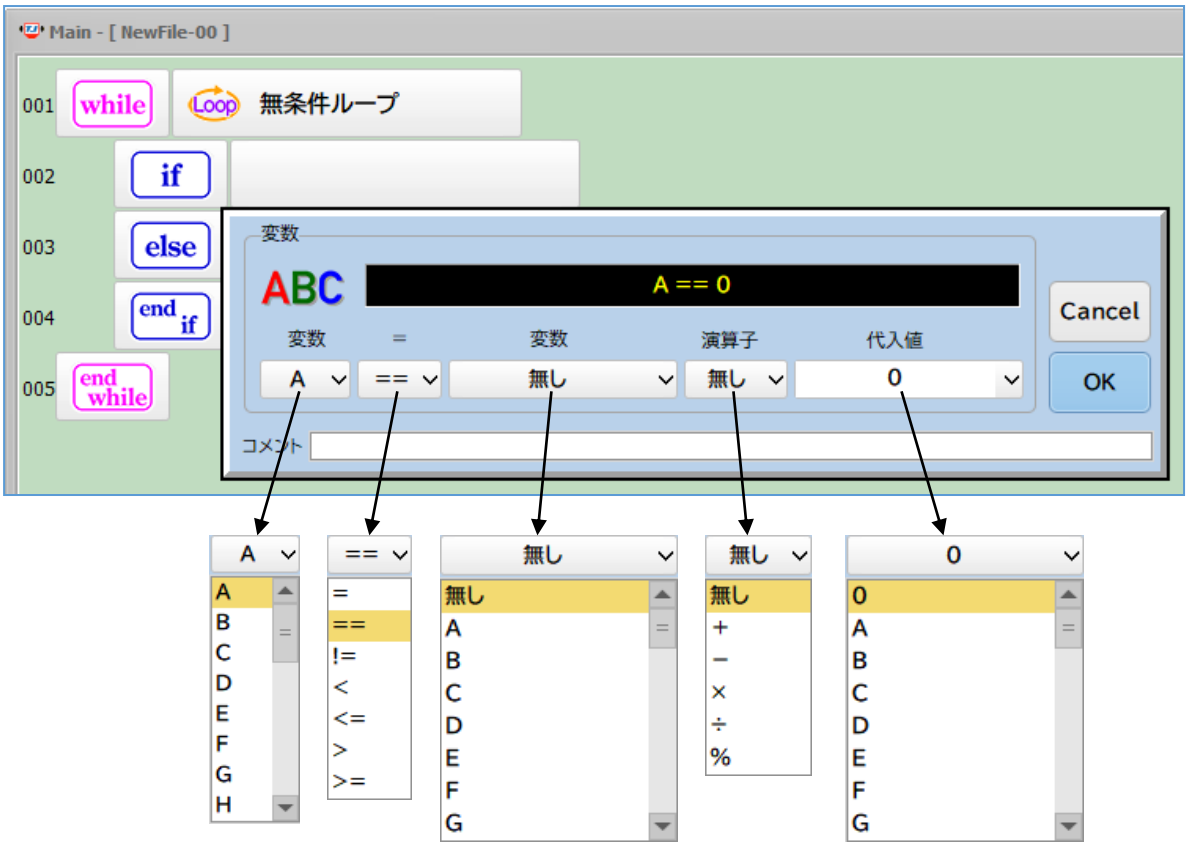
◇変数と比較する場合の注意

　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倍して変数Aに1500を代入します。

1-11. 変数チェック 

事前に代入された変数を選択して条件判定を行います。



変数は、A～Z の 26 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 CN10 に接続されたセンサーの値も変数へ代入し比較することが出来ます。

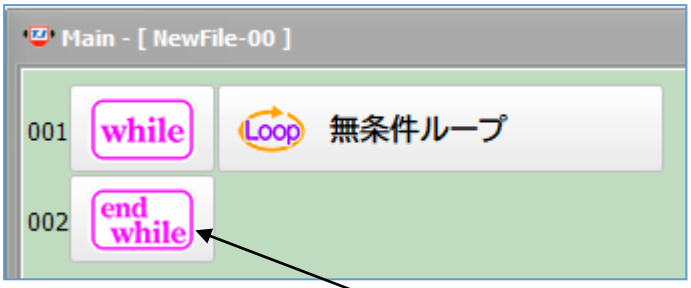
ボタン名称の例

“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 より大きき場合
“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、コピーと貼付け

2-1. ボタンの挿入

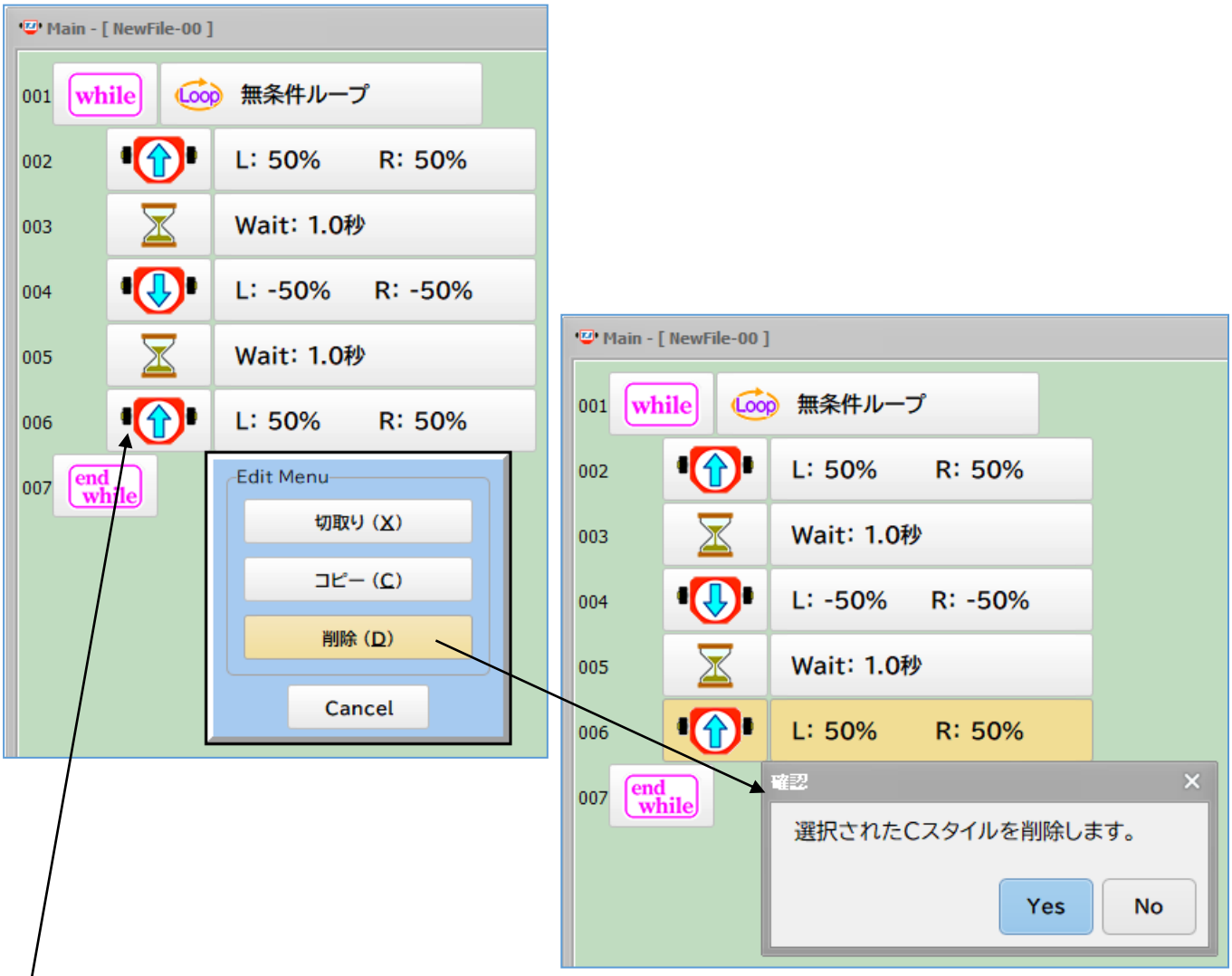
① 先に、挿入したいプログラムボタンを選択しておきます。



② 次に、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。

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



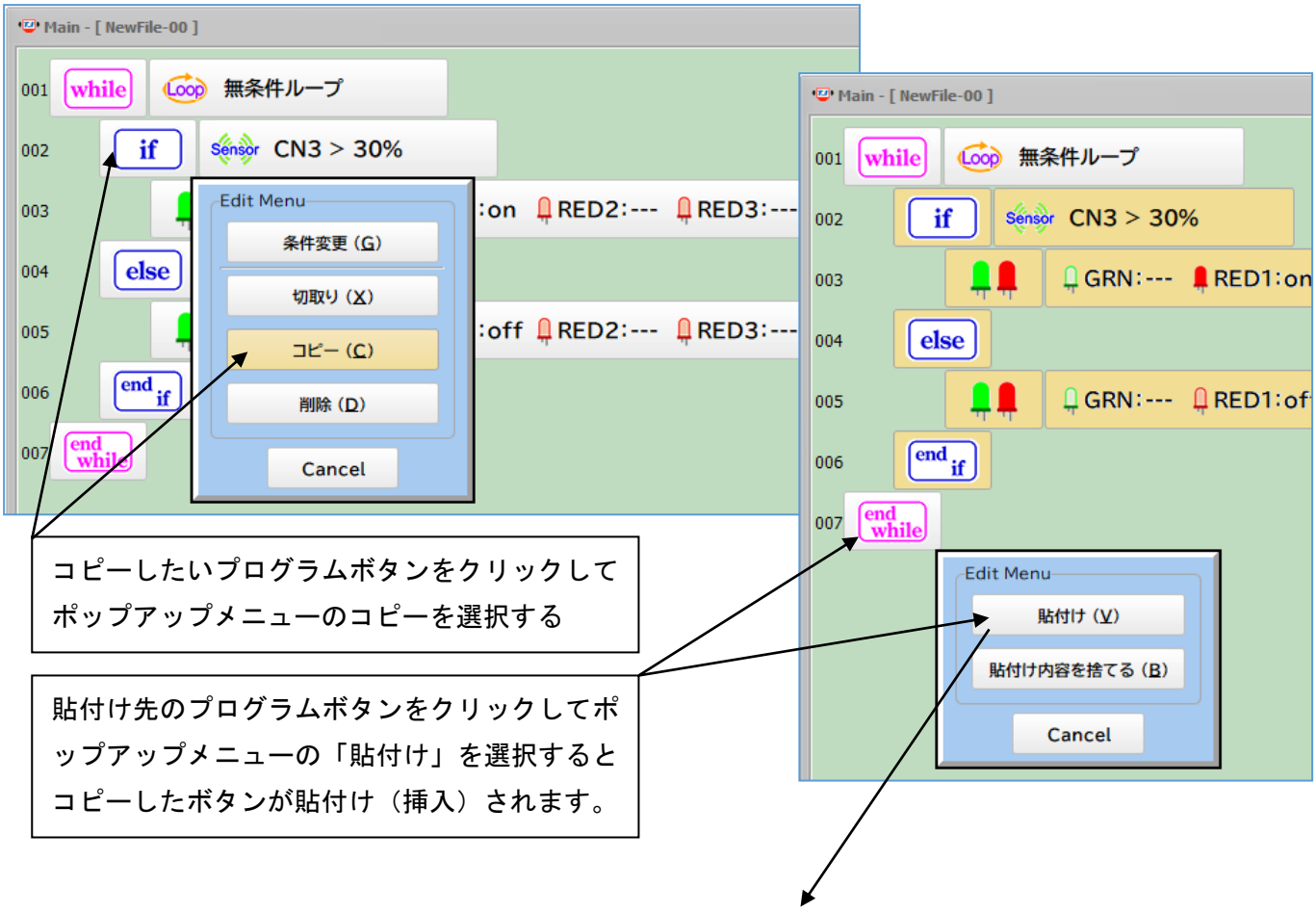
① 削除したい行のプログラムボタンをクリックするとポップアップメニューが表示されますので、「削除」を選択します。

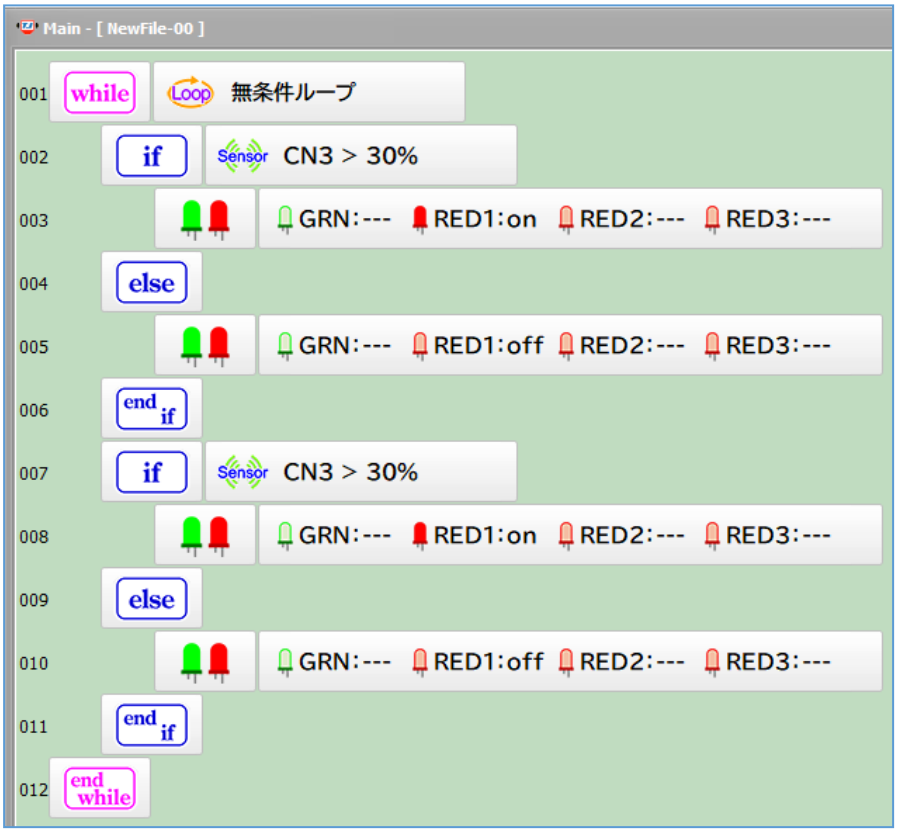
② ボタン削除の確認ダイアログが表示されますので、「はい」をクリックすると、その行のボタンが削除され、行が詰められます。

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





コピーされた 002～006 が 007～011 に貼付けられました。

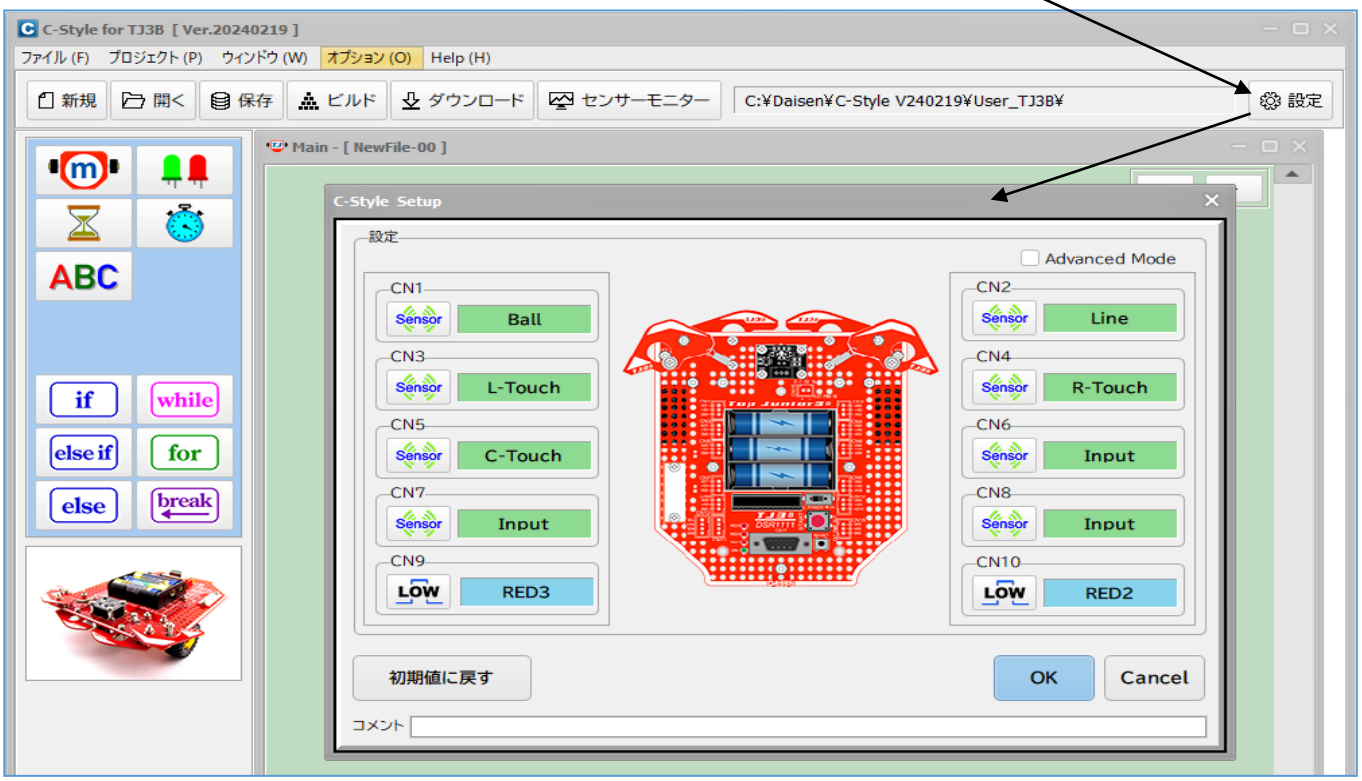
コピーを連続して行うと、以前にコピーした内容の後に追加され、貼付け時に全て挿入されます。
この機能は、切取りにおいても同じ働きをします。
コピーや切取りで蓄積された貼付け内容が不要なった場合は、編集領域でクリックしてポップアップメニューを表示させて、「貼付け内容を捨てる」を選択します。

３． 入出力設定

TJ3B は CN1～CN8 のアナログ入力 CN9 と CN10 はデジタル出力の設定になっています。
出荷時は、CN1:赤外線ボールセンサー、CN2:赤外線反射センサー(ラインセンサー)、CN3～CN5:左右と中央のタッチセンサーに接続され、CN6～CN8 は予備のアナログ入力の設定で、CN9～CN10 は RED3 と RED2 の LED 出力となっています。
Setup（設定）の画面から CN1～CN10 はアナログ入力やデジタル出力の設定に変えることが出来ます。
さらにオプションパーツを接続することにより CN1～CN10：サーミスター温度計入力、CN6:メロディーブザー制御、CN6～CN10:サーボモーター制御、CN7～CN10 は超音波距離センサー入力が行えます。
また ☑Advanced Mode にすると I2C 通信接続による各種オプションパーツの入出力制御や UART-DSUB9 通信によるデータロギング機能が行えます。

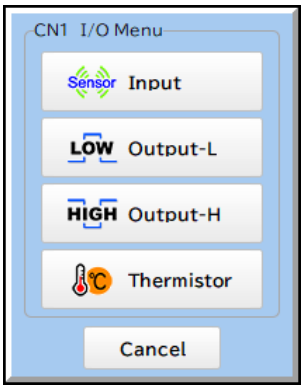
3-1. Setup ボタンの表示

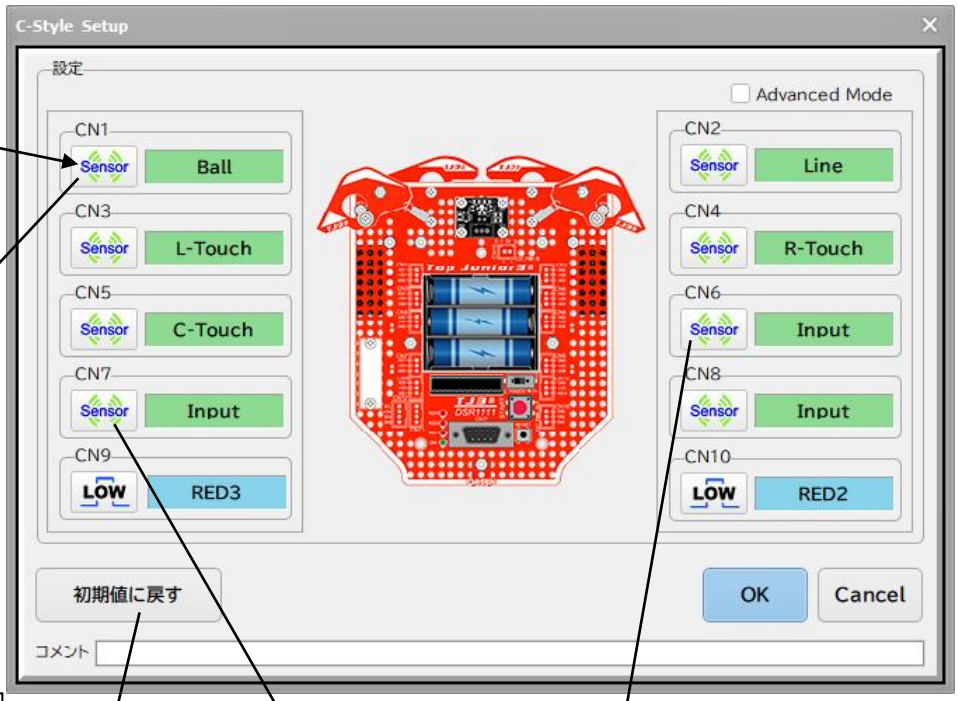
オプションメニューの「設定ボタンを表示」を選択しますと、画面右側に 設定 ボタンが表示されます。
初めてボタン表示選択をした場合は、入出力の設定が行える「設定(Setup)」ダイアログが表示されます。



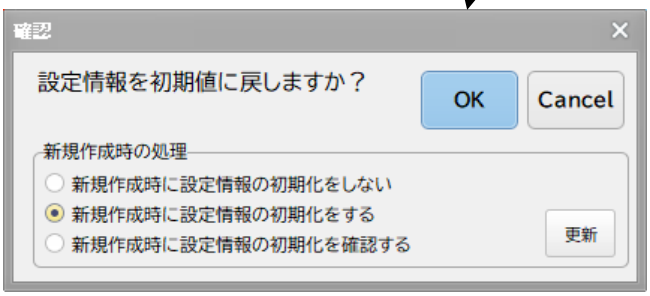
3-2. Setup（入出力設定）

Sensor 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。

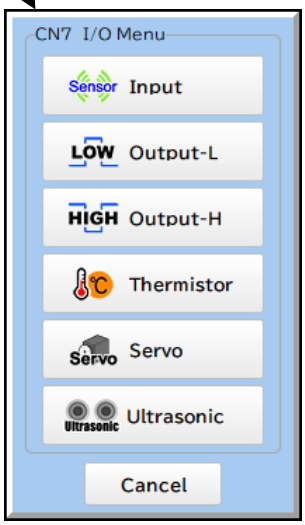




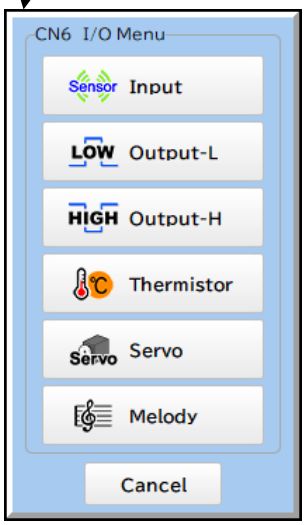
CN1～CN5 は
アナログ入力、デジタル出力、
サーミスター温度入力が選択
出来ます。



［ 初期値に戻す ］ボタンは全ての入出力設定を初期の状態に戻します。
新規作成時の処理を選択し「更新」ボタンをクリックすると以後の新規作成時では選択された処理で初期化が実行されます。



CN7～CN10 は更に
サーボモーター制御、
超音波センサー入力が
選択出来ます。

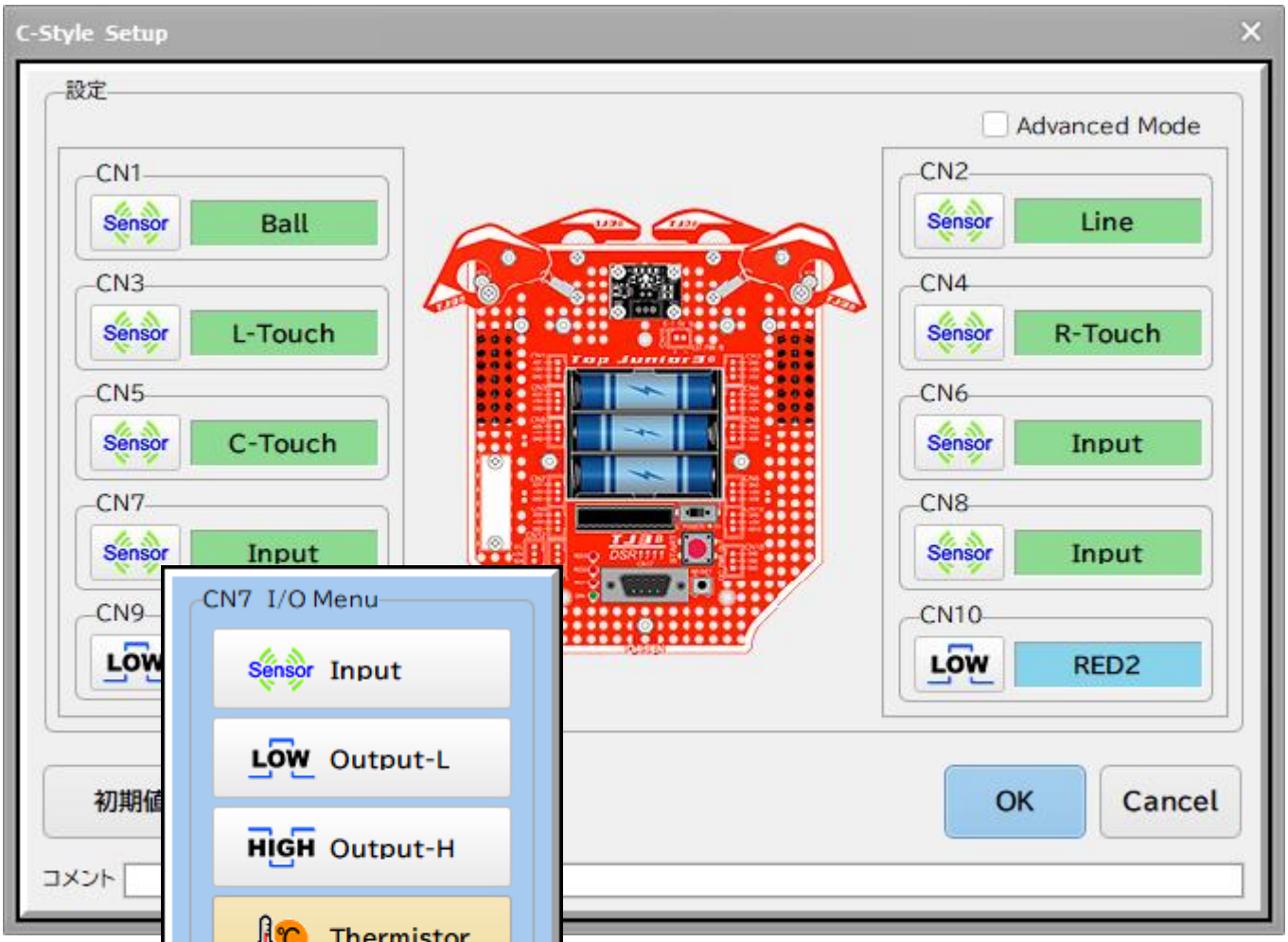


CN6 は更に
サーボモーター制御、
メロディーブザー制御
が選択出来ます。

※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ロボット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. サーミスター温度センサーを使う

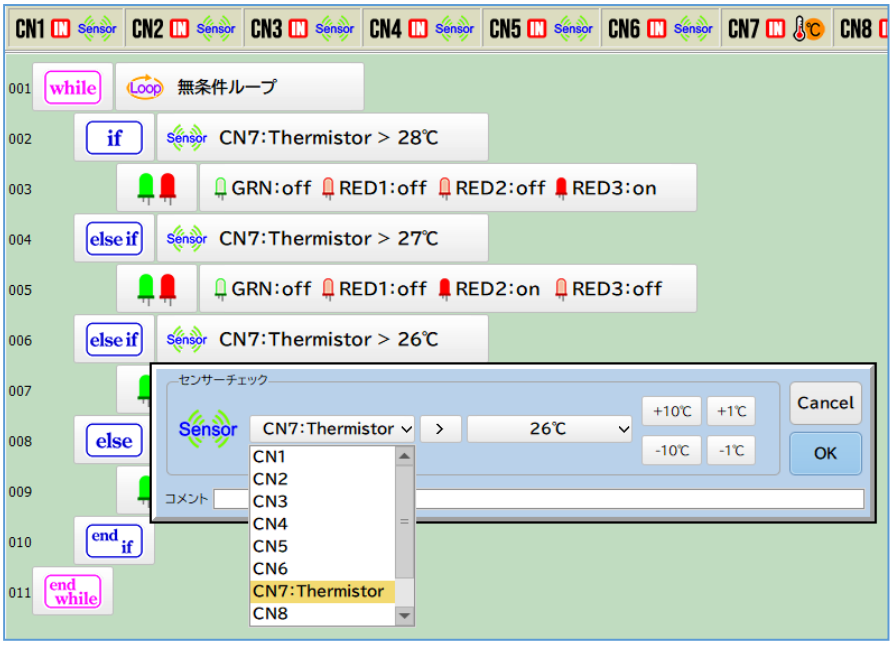
CN1～CN10 の Sensor をクリックして I/O Menu の ℃ Thermistor を選択します。



CN1～CN10 には別売の DSR1802 Thermistor 温度センサーを接続することが出来ます。

但し、CN1～CN5 には Ball, Line, Touch センサーが実装されていますので、それらを取外す必要があります。

サーミスター温度監視のプログラム例



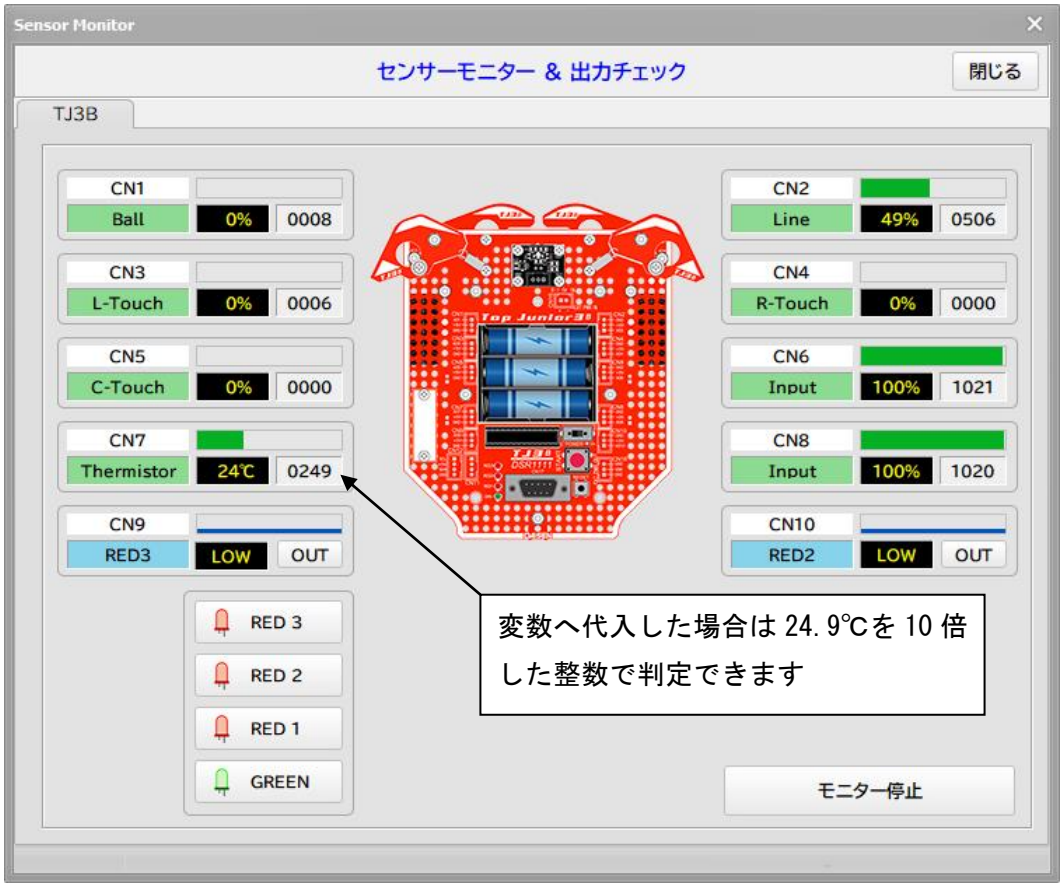
CN7 の温度が 28℃以上で
LED3 が点灯

CN7 の温度が 27℃以上で
LED2 が点灯

CN7 の温度が 26℃以上で
LED1 が点灯

それ以外の温度の場合は
GRN-LED が点灯

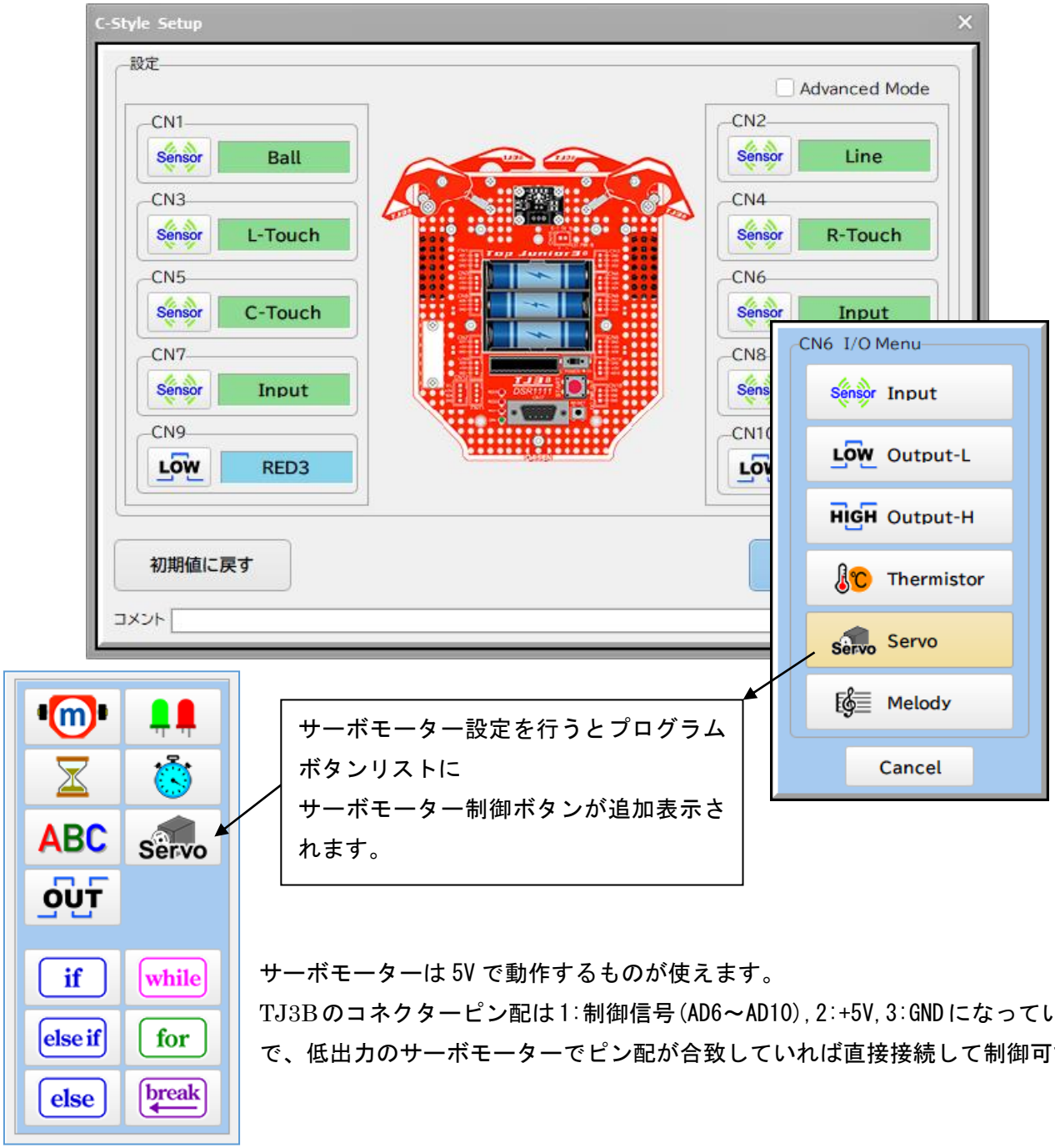
Thermistor 設定時のセンサーモニター画面



※センサーモニターで動作確認を行うには、該当する CN1～CN10 に Thermistor の設定をしたプログラムをダウンロードする必要があります。

3-4. サーボモーターを使う

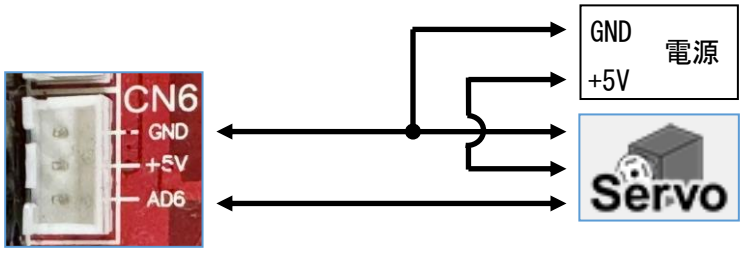
CN7～CN10 の Sensor をクリックして I/O Menu の Servo を選択します。



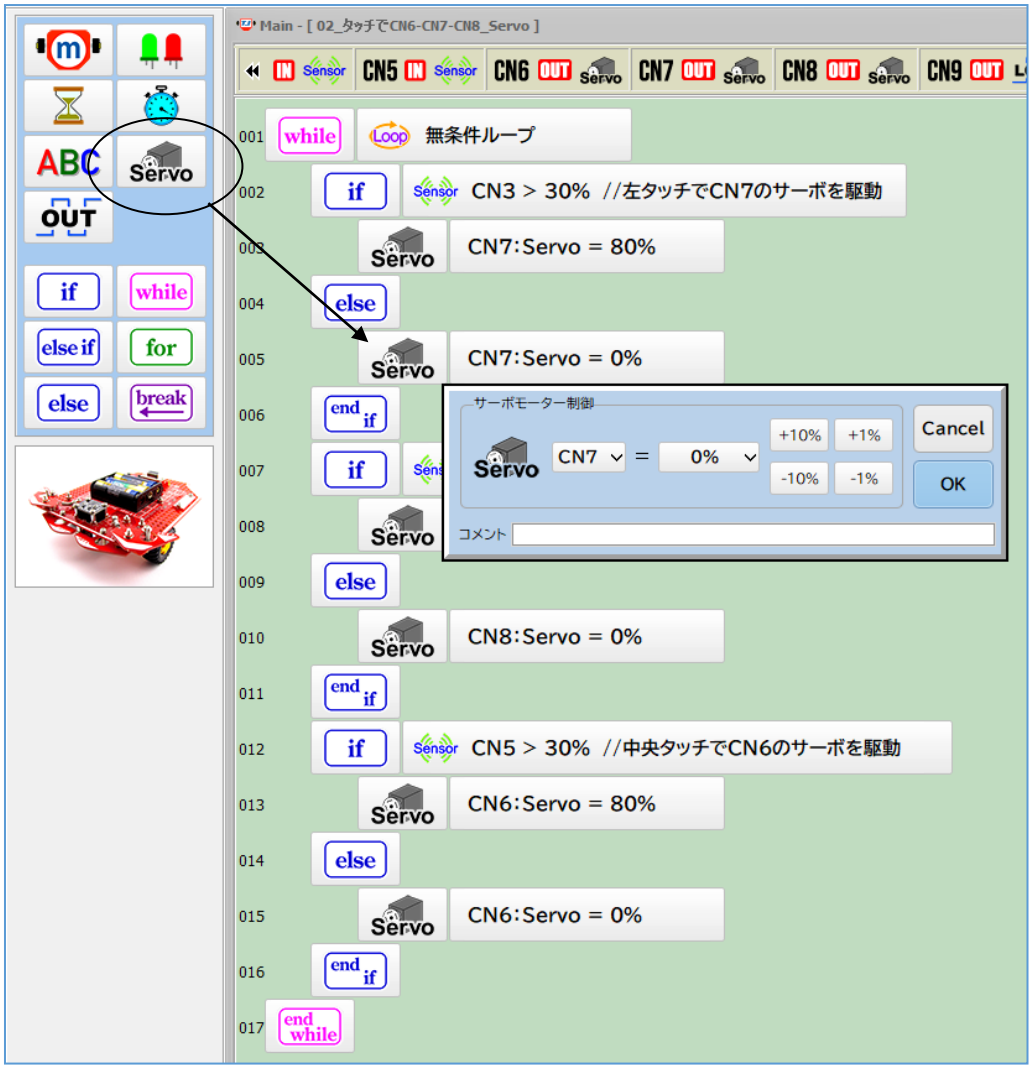
サーボモーター設定を行うとプログラムボタンリストに
サーボモーター制御ボタンが追加表示されます。

サーボモーターは 5V で動作するものが使えます。
TJ3B のコネクターピン配は 1:制御信号(AD6～AD10), 2:+5V, 3:GND になっていますので、低出力のサーボモーターでピン配が合致していれば直接接続して制御可能です。

高出力のサーボモーターの場合、電源線はロボットのコネクタに接続しないで、別の電源から直接サーボモーターに接続して下さい。ロボットからの電源供給でロボットの電源回路が破損する場合があります。



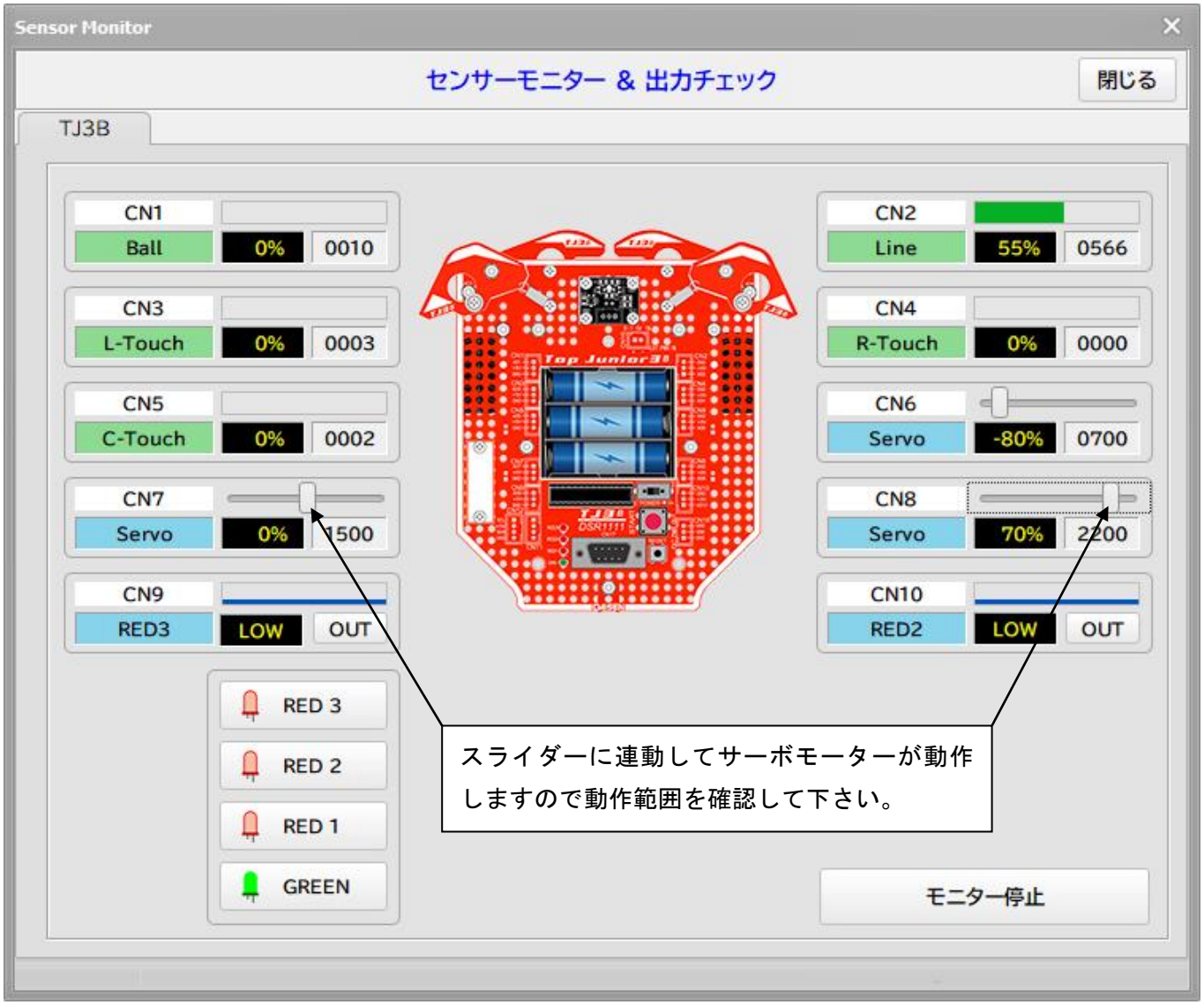
サーボモーター制御プログラム例



この例では、左右と中央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 -100%（500μ秒）～ +100%（2500μ秒）で設定できます。
中央の位置が 0%（1500μ秒）です。

使用するサーボモーターによっては、最小値(-100%)や最大値(+100%)が限界を超えている場合があります。
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下さい。

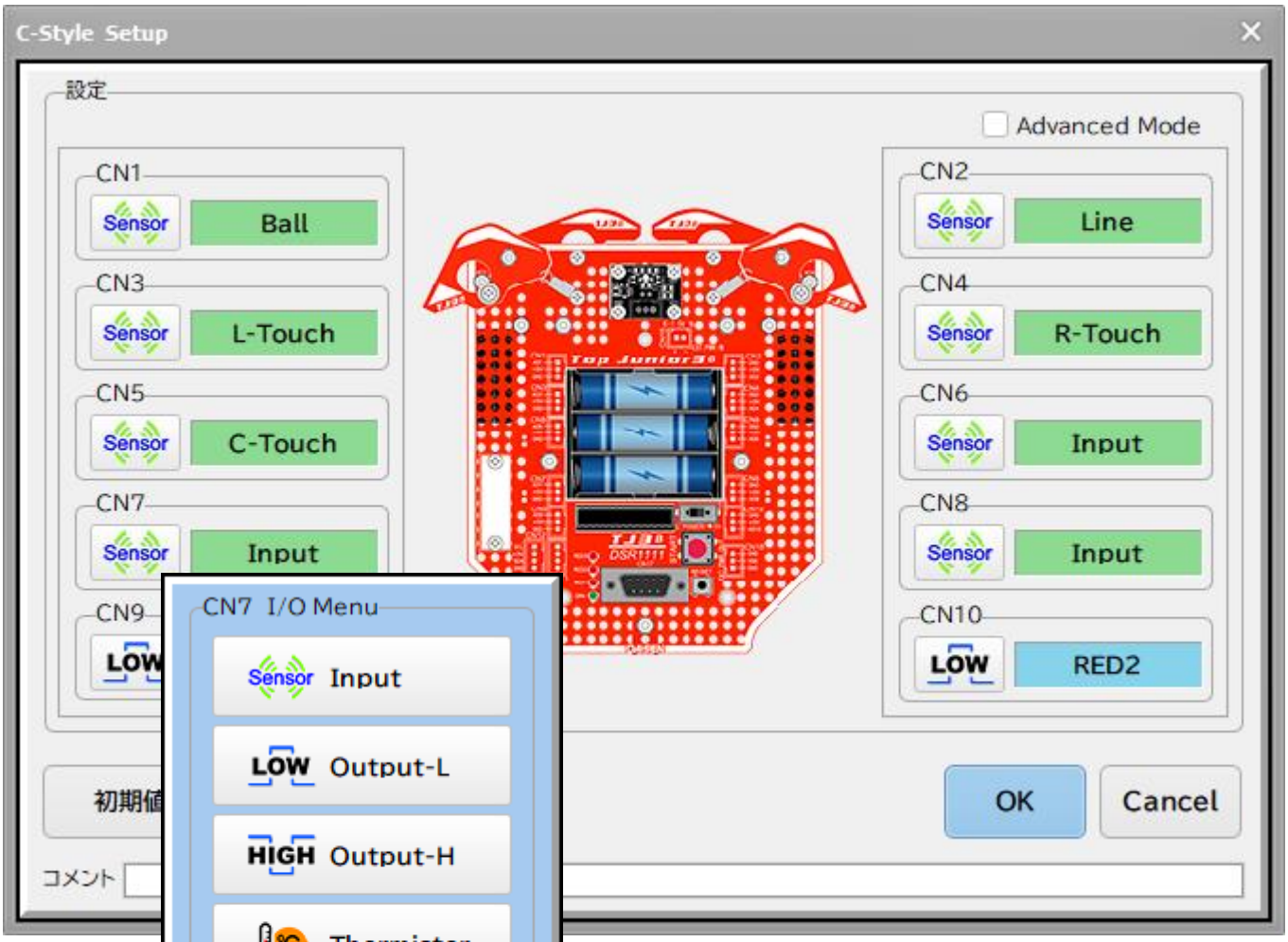
サーボモーター設定時のセンサーモニター画面



※センサーモニターで動作確認を行うには、該当する CN6～CN10 にサーボモーターの設定をしたプログラムをダウンロードする必要があります。

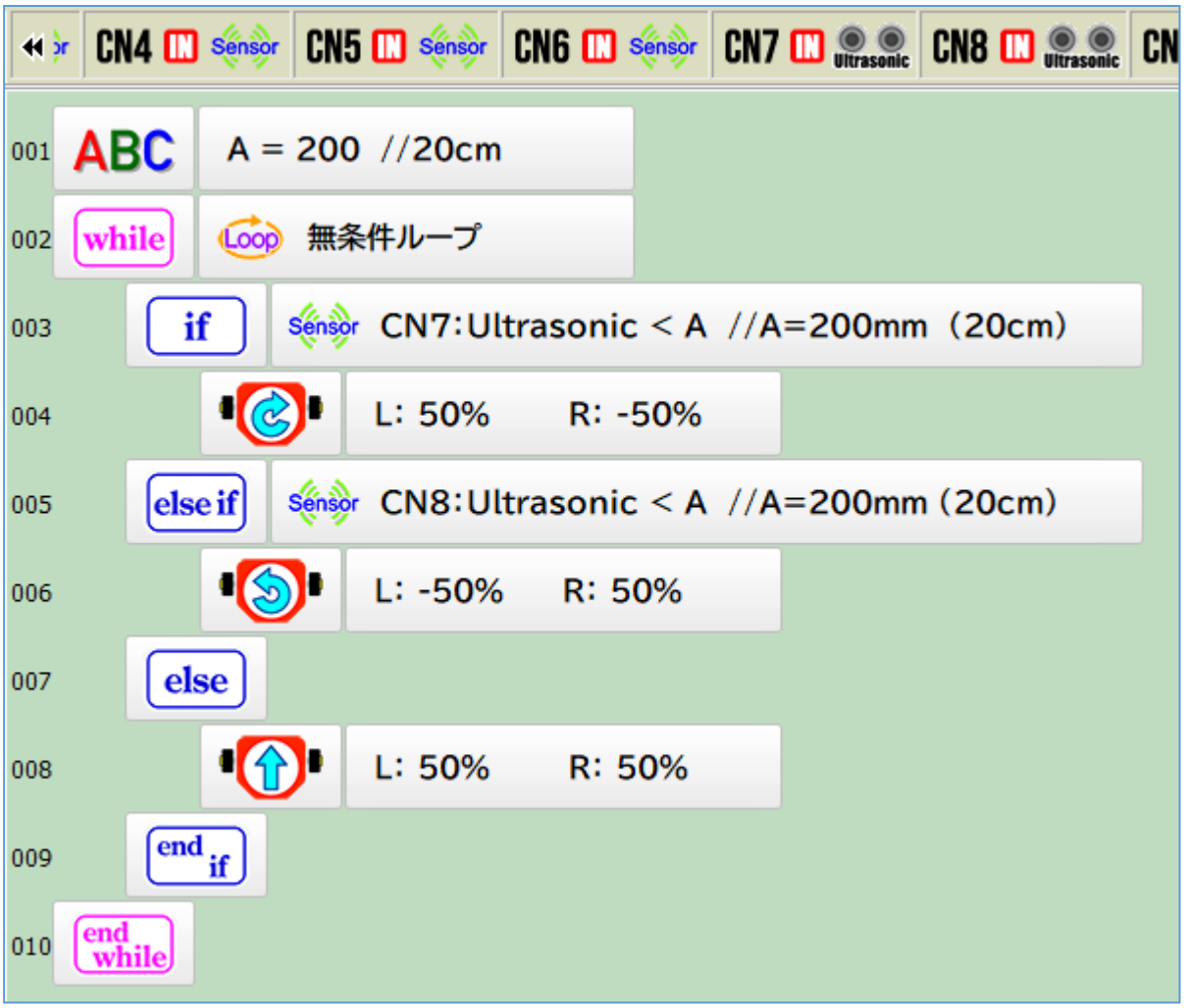
3-5. 超音波距離センサーを使う

CN7～CN10 の Sensor をクリックして I/O Menu の Ultrasonic を選択します。



CN7～CN10 には別売の Parallax 製の PING 超音波距離センサーを接続します。

CN7 と CN8 に超音波距離センサーを接続して、左右の壁の間を通り抜けるプログラム例



直接判定する場合は、cm の単位となります。

CN7 < 20cm

CN8 < 20cm

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 = 200 (200mm)

CN7 < 変数 A (CN7 < 20cm と同じ内容となります)

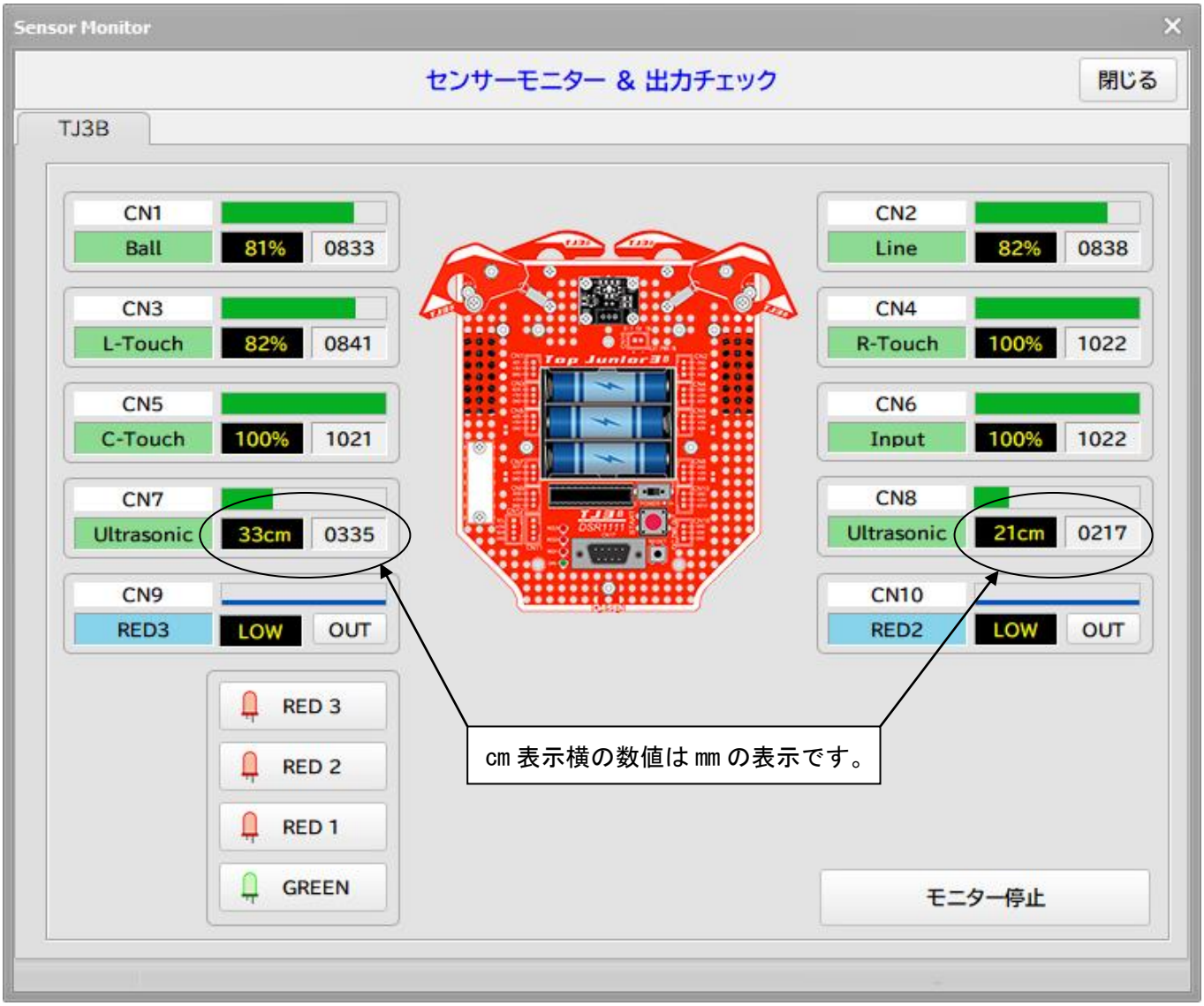
CN8 < 変数 A (CN8 < 20cm と同じ内容となります)

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 = CN8 (変数 A の値は mm の単位となります)

変数 A < 200 (CN8 < 20cm と同じ内容となります)

CN7,CN8 を超音波距離センサーに設定した時のセンサーモニター画面



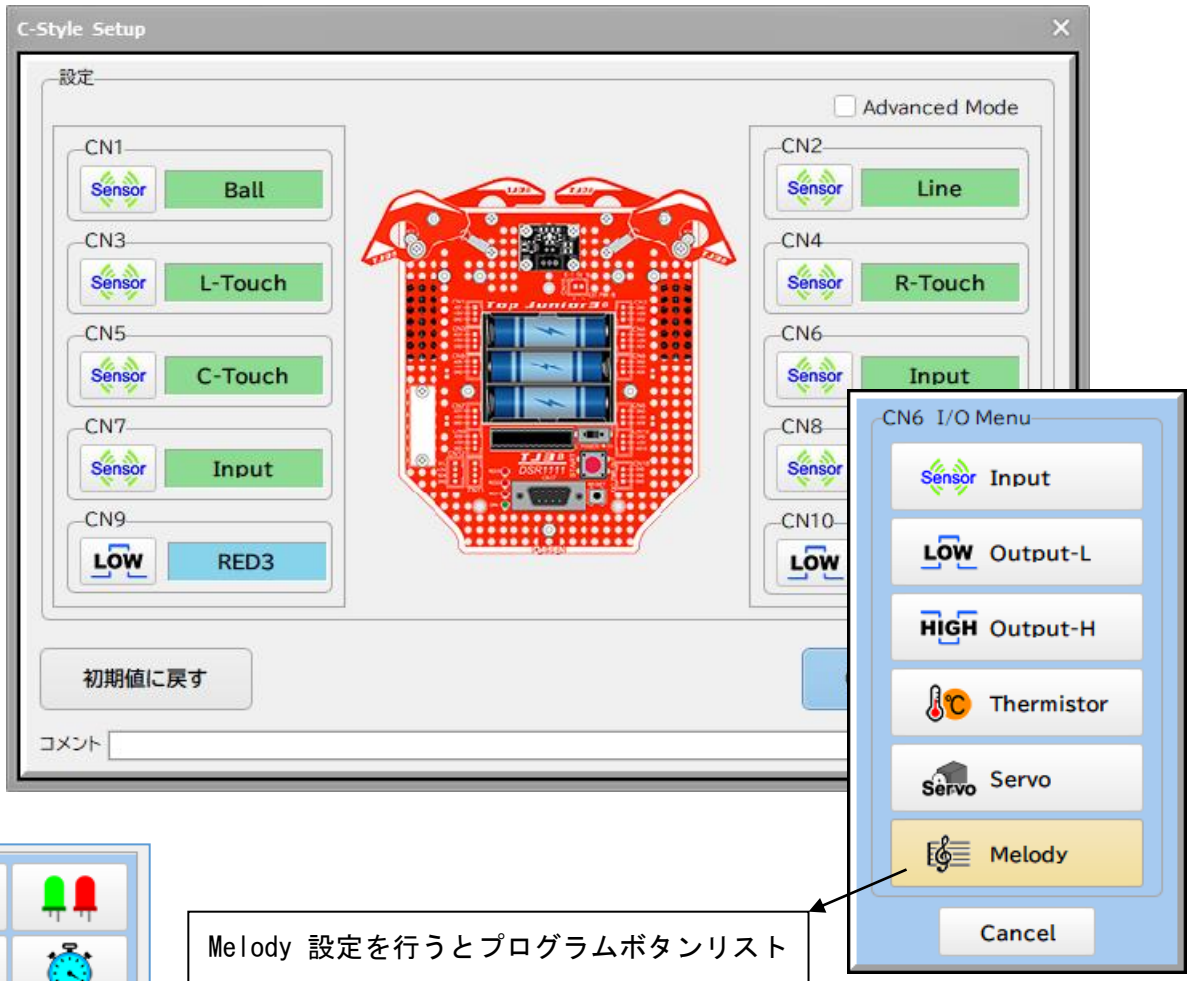
超音波距離センサーの設定がされている場合は %表示から cm 表示に変わります。

※CN9 と CN10 に超音波距離センサーを設定した場合、RED3 と RED2 の LED が予め接続されていますので LED の点灯制御を行っても超音波距離センサーが動作中は点滅しますので LED 制御は無効となります。センサーの計測には問題ありません。

※センサーモニターで動作確認を行うには、該当する CN7～CN10 に Ultrasonic(超音波距離センサー)の設定をしたプログラムをダウンロードする必要があります。

3-6. メロディーブザーを使う

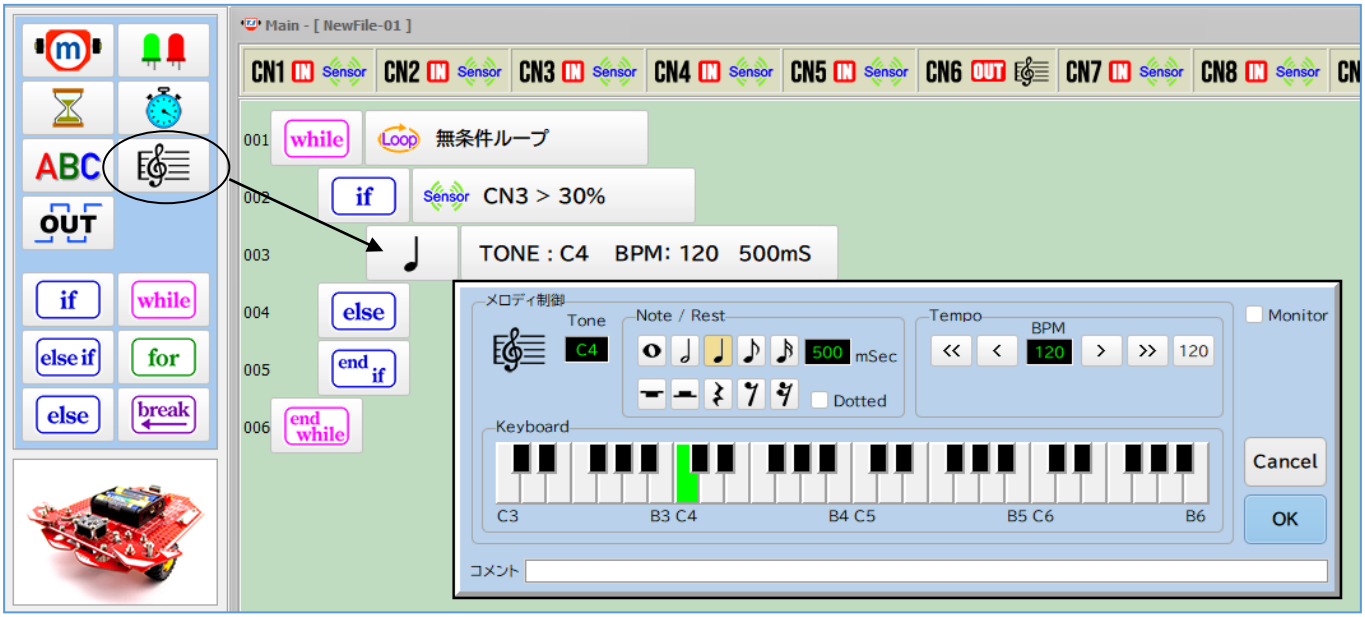
CN6 の Sensor をクリックして I/O Menu の Melody を選択します。



Melody 設定を行うとプログラムボタンリストにメロディー制御ボタンが追加表示されます。

CN6 には別売の Buzzer DSR1801 を接続します。

左タッチ(CN3)でメロディーブザーを鳴らすプログラム例



プログラムボタンリストから  メロディーボタンを選んで編集領域に置くとメロディー制御ダイアログが表示されます。
4 オクターブの音階とメロディー、休符や音階のテンポ等が編集できます。

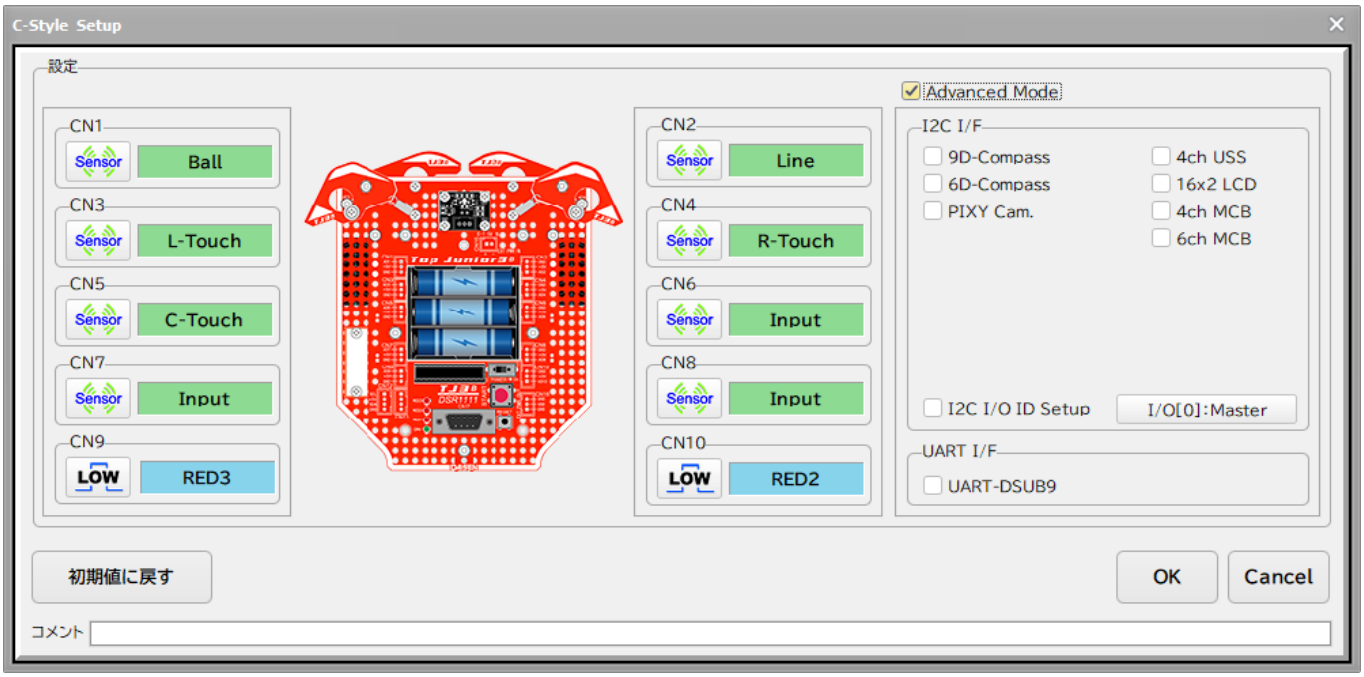
事前に CN6 がメロディー設定された任意のプログラムをダウンロードして TJ3B の電源は入ったまま通信ケーブルを接続している状態でダイアログ右上の☑ Monitor にすると音階の鍵盤をクリックするとメロディーを聞くことが出来ます。

４． 拡張機能の設定

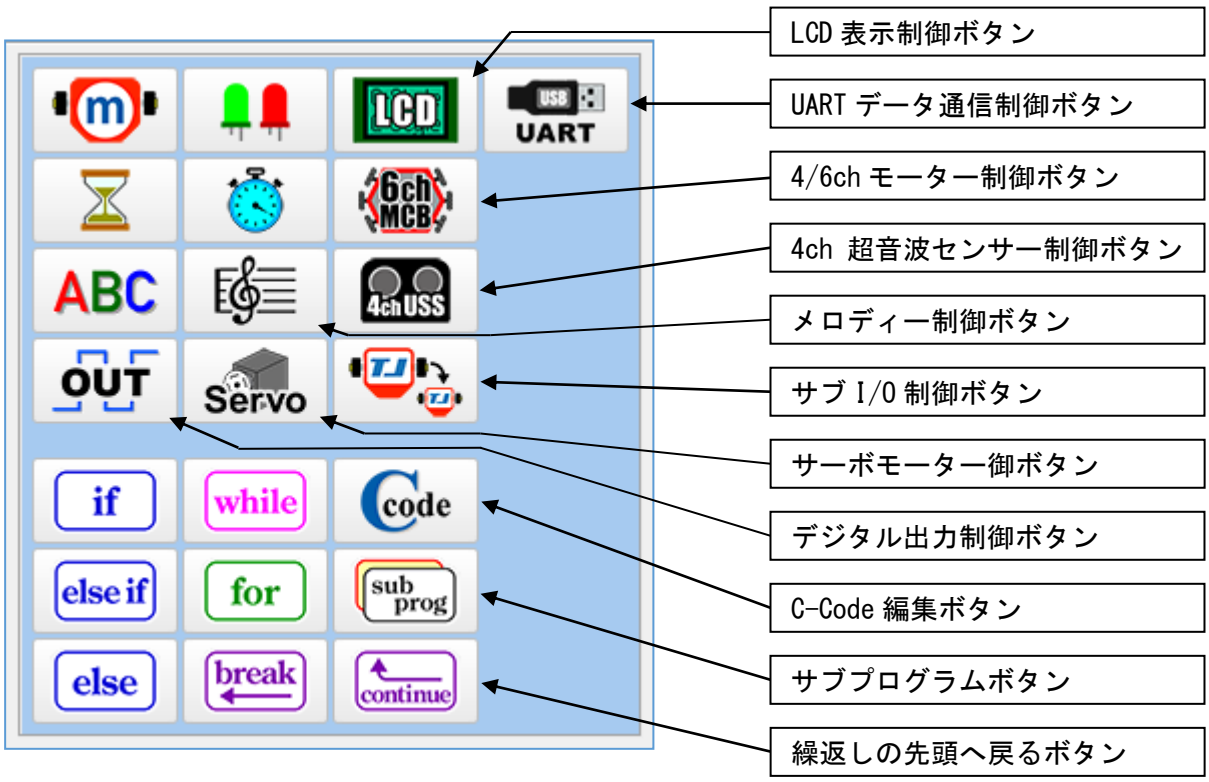
4-1. 拡張機能設定画面の表示

Setup 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 I2C 通信による拡張機能を設定するチェックボックスの一覧が表示されます。

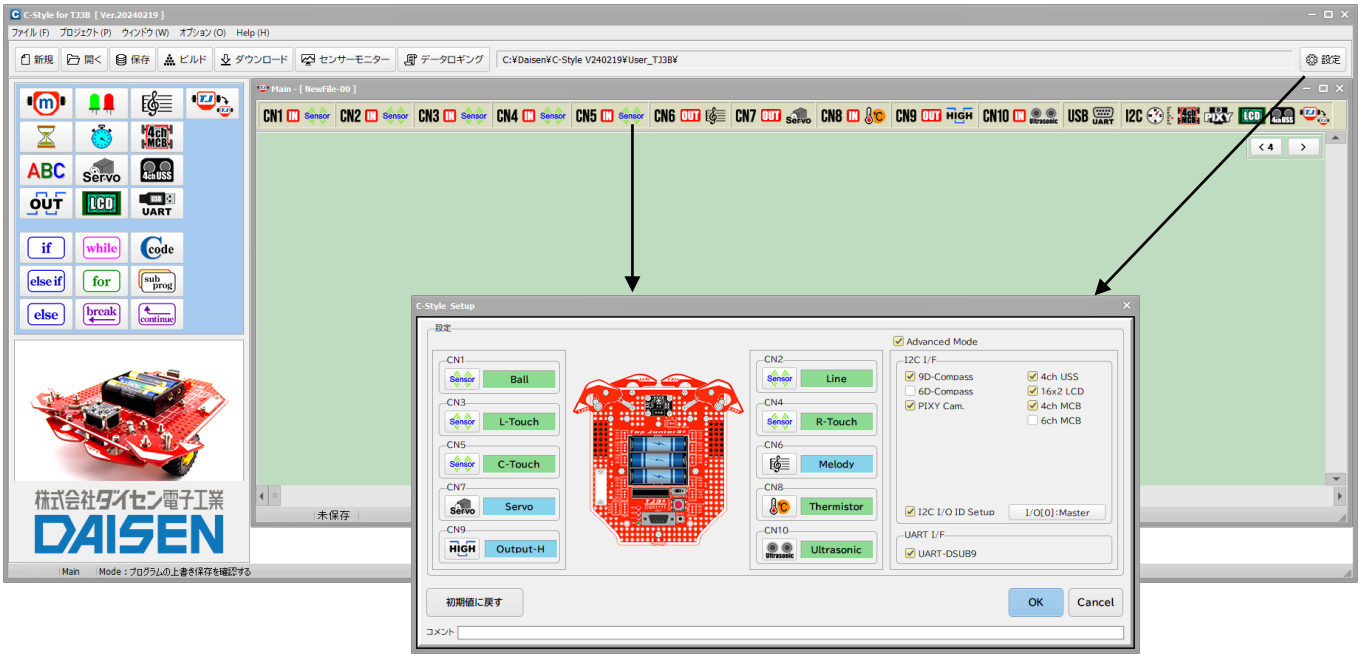
6D/9D-Compass(多機能電子コンパス：DSR1401/1603)、PixyCam.イメージセンサー、4ch USS(超音波距離センサーアダプター：DSR1608)、e-Gadget 用の 16x2 LCD 表示機、4ch/6ch モーターコントロールボード、最大 8 台までの TJ3B を拡張センサーボードとして使用する機能、TJ3B 本体に実装されているダウンロード用 DSUB9 のコネクタを介してデータロギング機能などの拡張機能の設定が出来ます。



拡張機能を設定するとプログラムボタンリストも設定に応じて拡張表示されます。



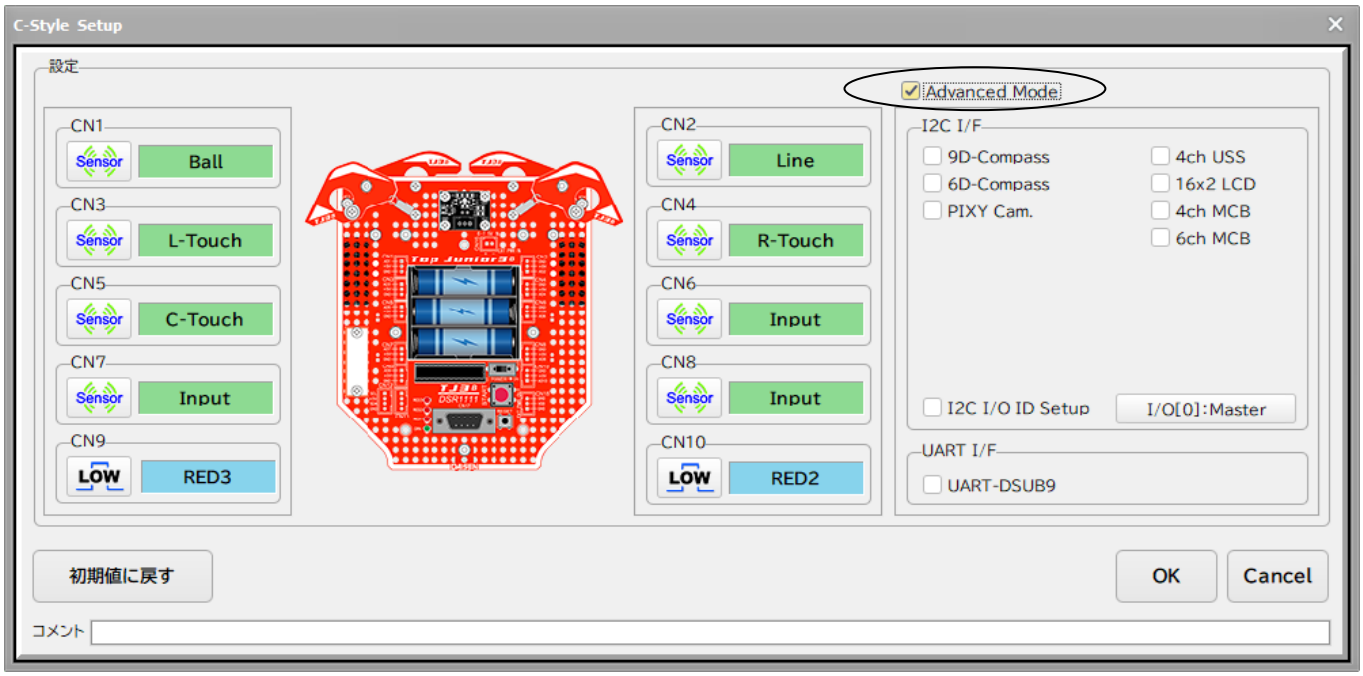
Setup 画面で OK ボタンをクリックして閉じると CN1～CN10 の設定状態と拡張機能の設定状態が編集画面上段に表示され合わせて画面右上に 設定 ボタンが表示されます。



設定 ボタン及び CN1～CN10 や拡張機能(I2C,UART)が表示されている場合それらをクリックすると再び Setup(入出力設定)ダイアログが表示されます。

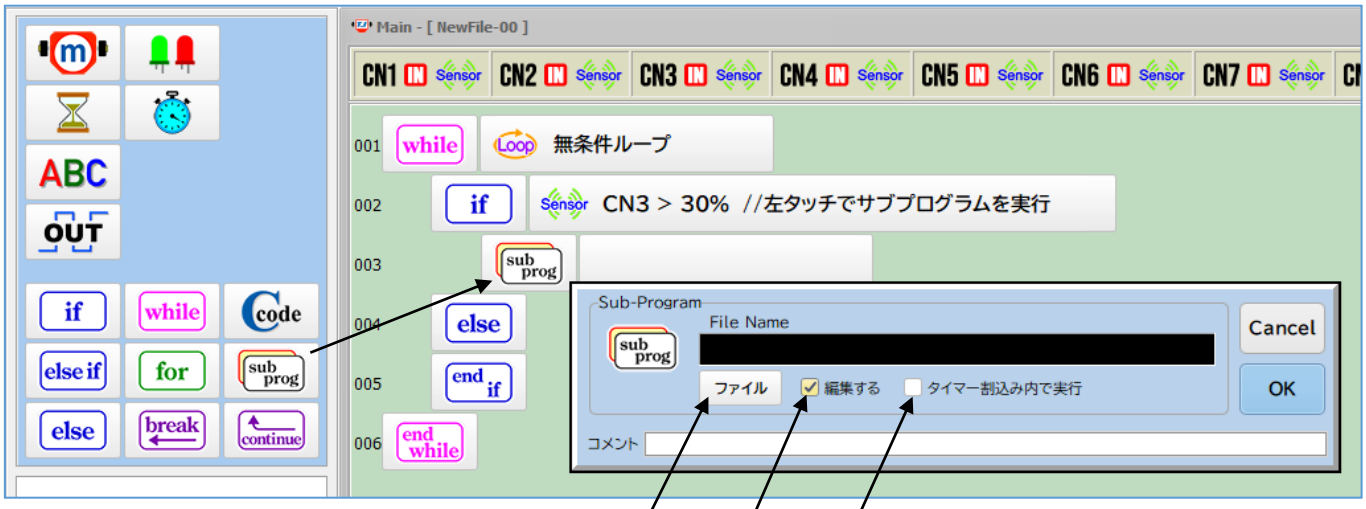
4-2. サブプログラムボタンの表示

Setup 画面で　☑**Advanced Mode**　にチェックを付けて「OK」ボタンで画面を閉じます。



Advanced Mode にするとプログラムボタンリストに 

ボタンが表示されます。
このボタンを使ってサブプログラムを最大 30 個まで作成することが出来ます。
同じ処理を何回も再利用したい場合や、メイン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出来ます。（サブルーチンと言います）

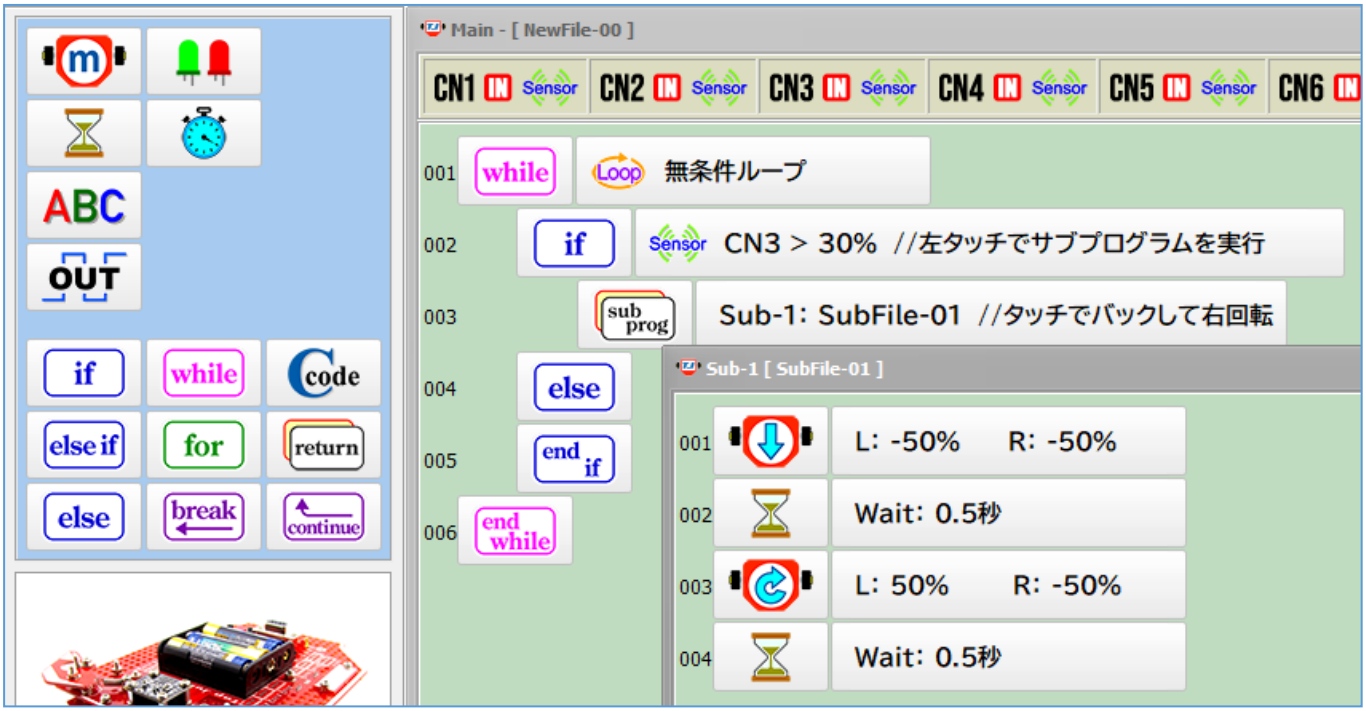


既に作成されたプログラムファイルを選択する場合は「ファイル」ボタンをクリックします

編集しない場合は「☐ 編集する」のチェックを外します

これにチェックを付けると特別なサブプログラムとなります。
「4-4. タイマー割込み内で実行するサブプログラム」を参照

4-3. 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。

■プログラムの保存

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ー内に保存して下さい。

■プログラムのビルド

ビルドボタンをクリックすると未保存のファイルは順次保存の確認を行います。ファイル名が未定の場合はこの時にファイル名を確定して下さい。キャンセルした場合はビルドを中止します。

■事前にサブプログラムを作成

メインプログラムで編集したファイルをサブプログラムとして後で使用する場合は、Setup 画面で設定する入出力設定及び Advanced Mode での I2C の設定はサブで使用しない場合でも将来メインとなる設定と同様の設定にして保存して下さい。メインとサブで異なる設定をした場合、ビルド失敗やビルドが成功しても正しく動作しません。

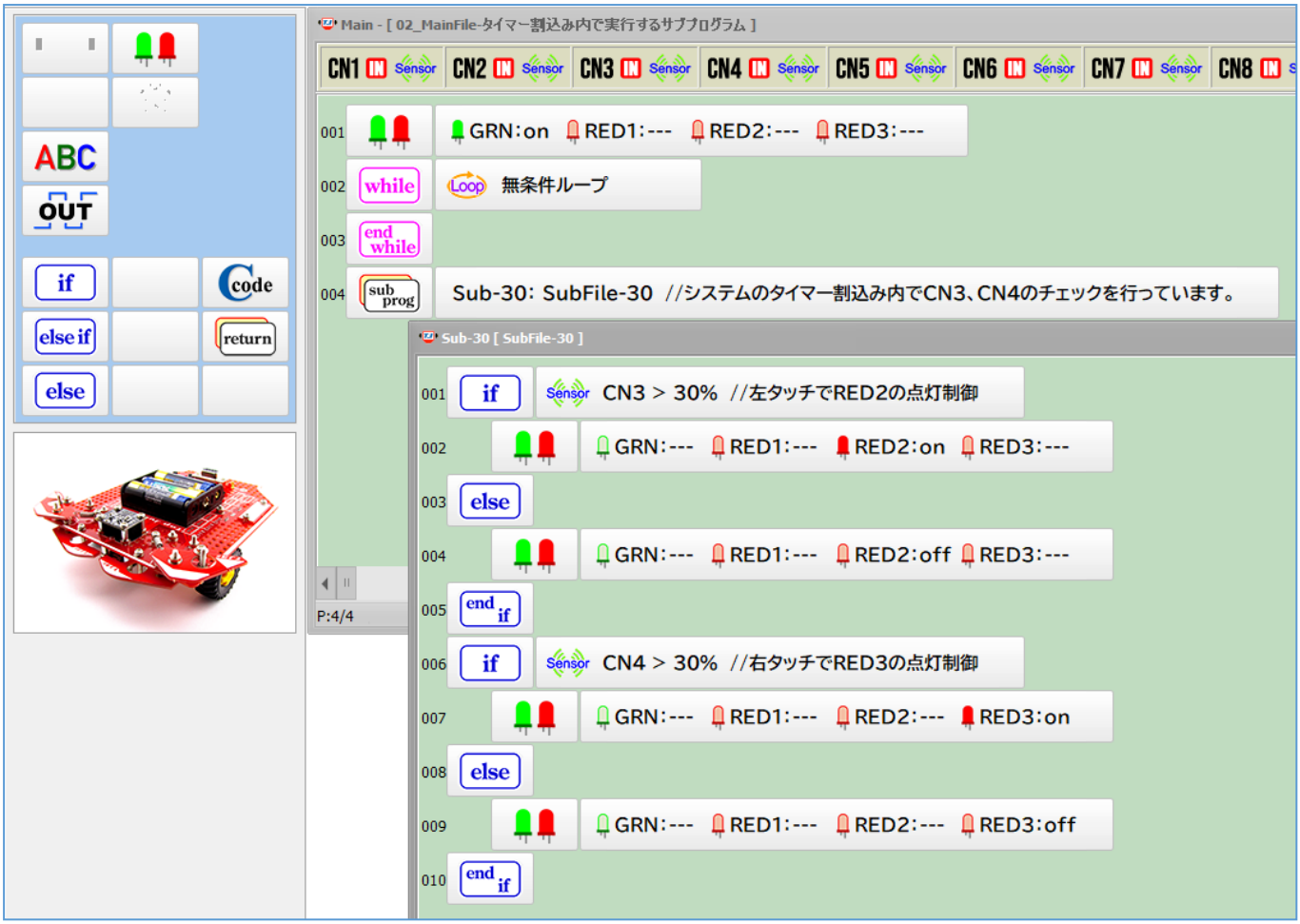
4-4. タイマー割込み内で実行するサブプログラム

サブプログラムダイアログで ☑ **タイマー割込み内で実行** にチェックを付けたサブプログラムは ”Sub-30” と表記され、TJ3B ファームウェアーの 1mS タイマー割込み内で実行する特別なサブプログラムとなります。

この場合のサブプログラムボタンはメインプログラムで実行されない場所に配置して下さい。

上記の例では、無条件ループの外側に配置されていますので、通常のサブプログラムでは実行されることはありませんが、タイマー割込み内で実行するサブプログラムになっている場合はシステムのタイマー割込み内に配置され 1mS 毎に発生する割込みで実行されることになります。

タイマー割込み内で実行するサブプログラム編集



この例では、サブプログラムが無条件ループの外側に置かれているので、通常実行されないのですが、実際はタイマー割込み内で常に実行されているので、CN3、CN4 のタッチチェックにより LED の点灯制御が行われます。

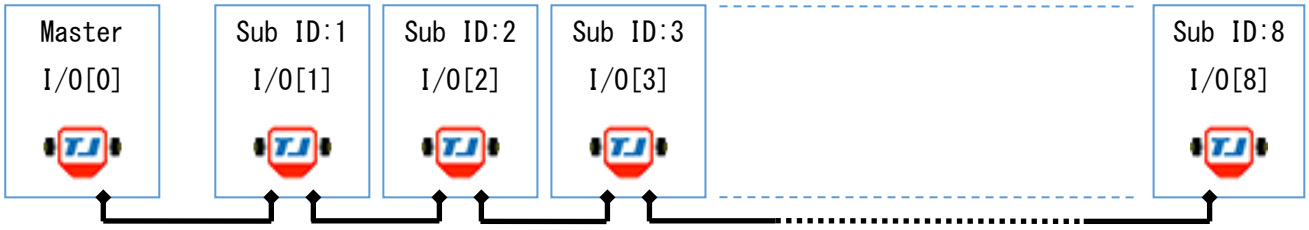
タイマー割込み内で実行するサブプログラムでは、変数演算、CN1～CN10 に接続されたアナログセンサーの判定、LED の点灯制御などが使用可能です。

※ご注意
1mS 毎の割込み内で実行する為、処理時間のかかるプログラムやモーター制御、超音波距離センサー、サーボモーター、I2C 等の他の割込み処理を必要とする命令は置くことが出来ません。

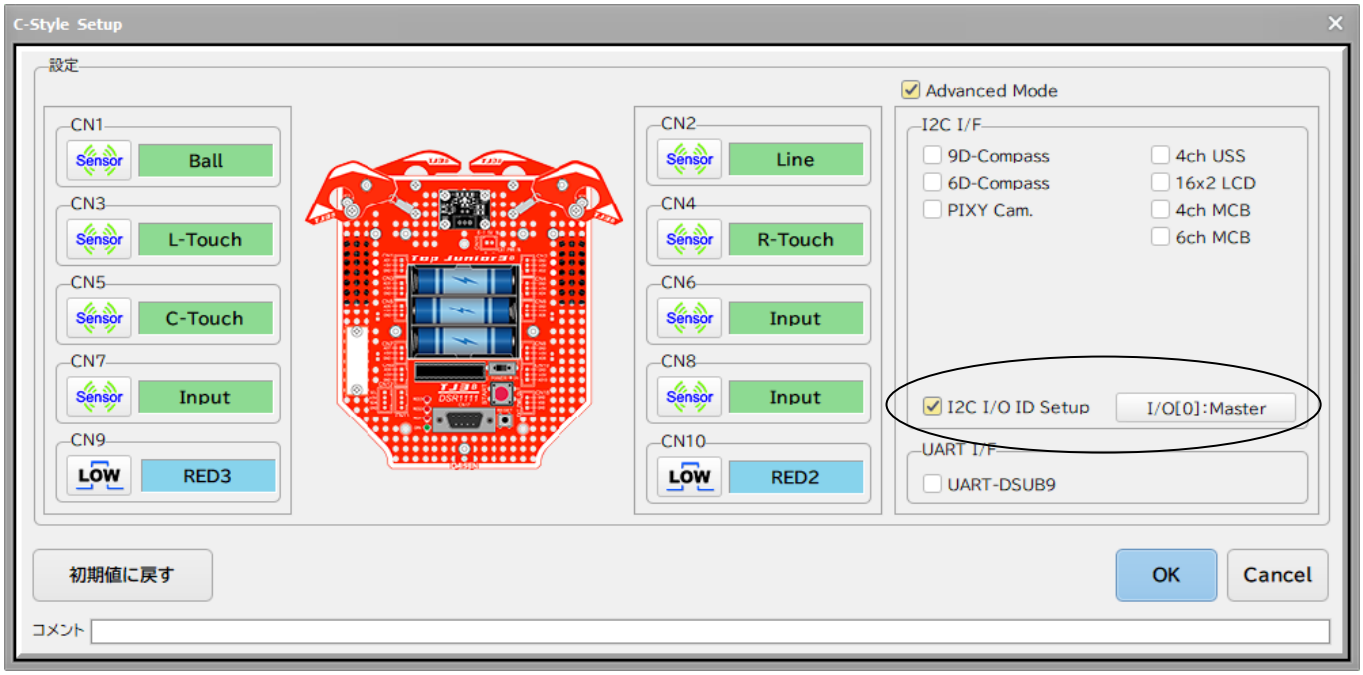
4-5. サブ I/O 制御（複数台の TJ3B を接続）

TJ3B は 3PIN の入出力コネクタが 10 個(CN1～CN10)ありますが、その他に 4PIN の I2C 通信仕様のコネクター(CN11,CN12)があります。この I2C 用コネクタを使って TJ3B を最大 8 台まで接続して親の TJ3B がセンサー情報や制御をコントールすることが出来ます。

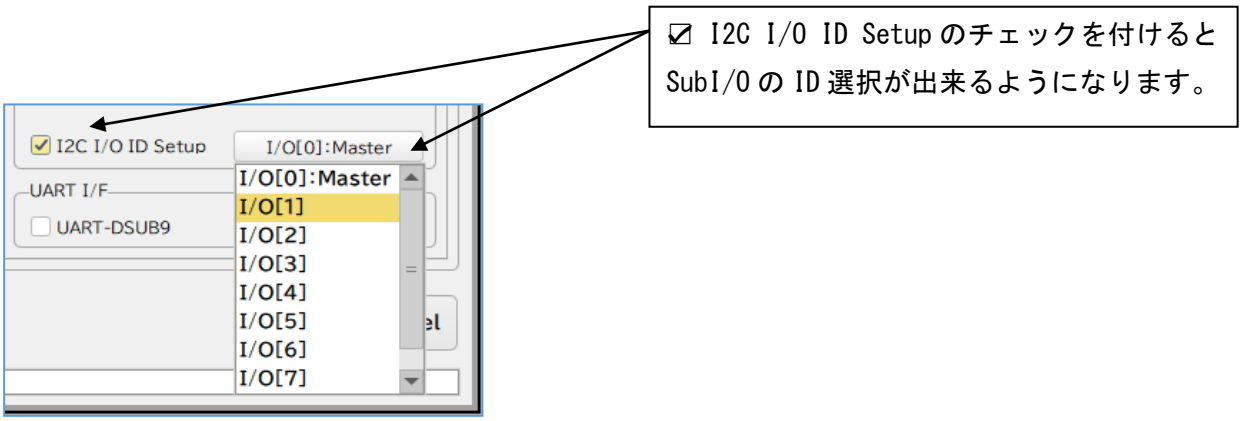
センサー情報を集約して管理する親の TJ3B のことを Master(I/O[0])と呼び、センサー情報を親へ引き渡す子の TJ3B のことを SubI/O (I/O[1]～I/O[8])と呼びます。



この設定を行うには、Setup 画面の ☑Advanced Mode にして I2C I/F 一覧で ☑I2C I/O ID Setup にチェックを付けます。

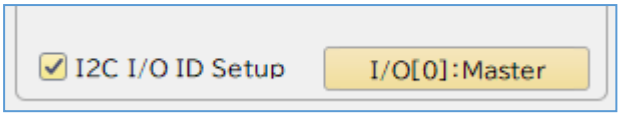


プログラム編集は、この Setup 画面で Master の ID:0 か SubI/O の ID:1 から 8 を決めてから行います。

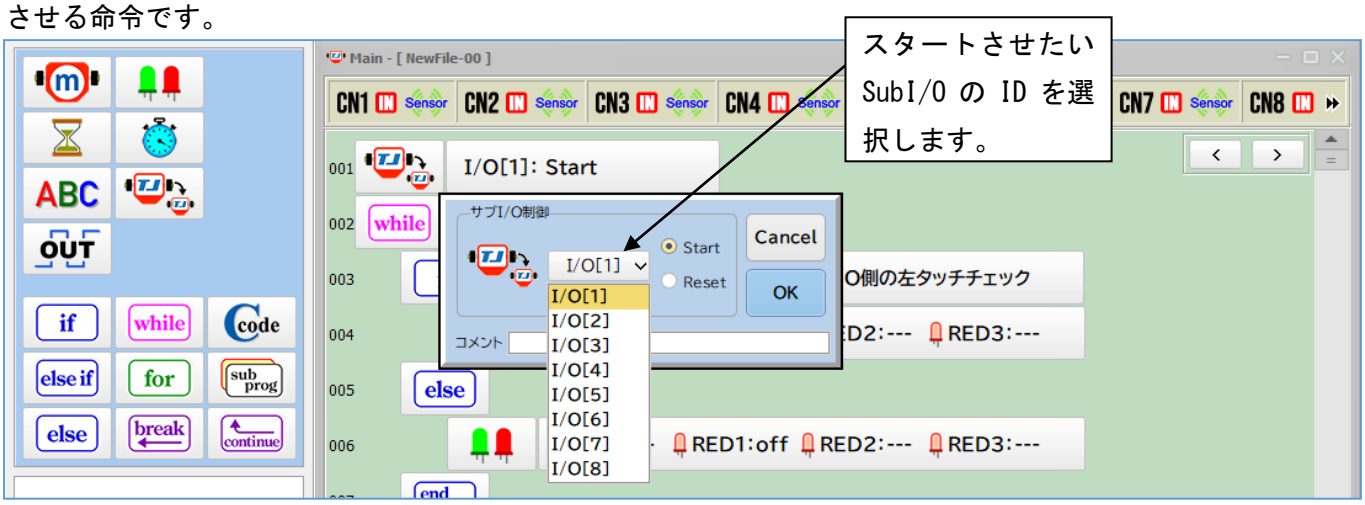


■Master 側のプログラム編集

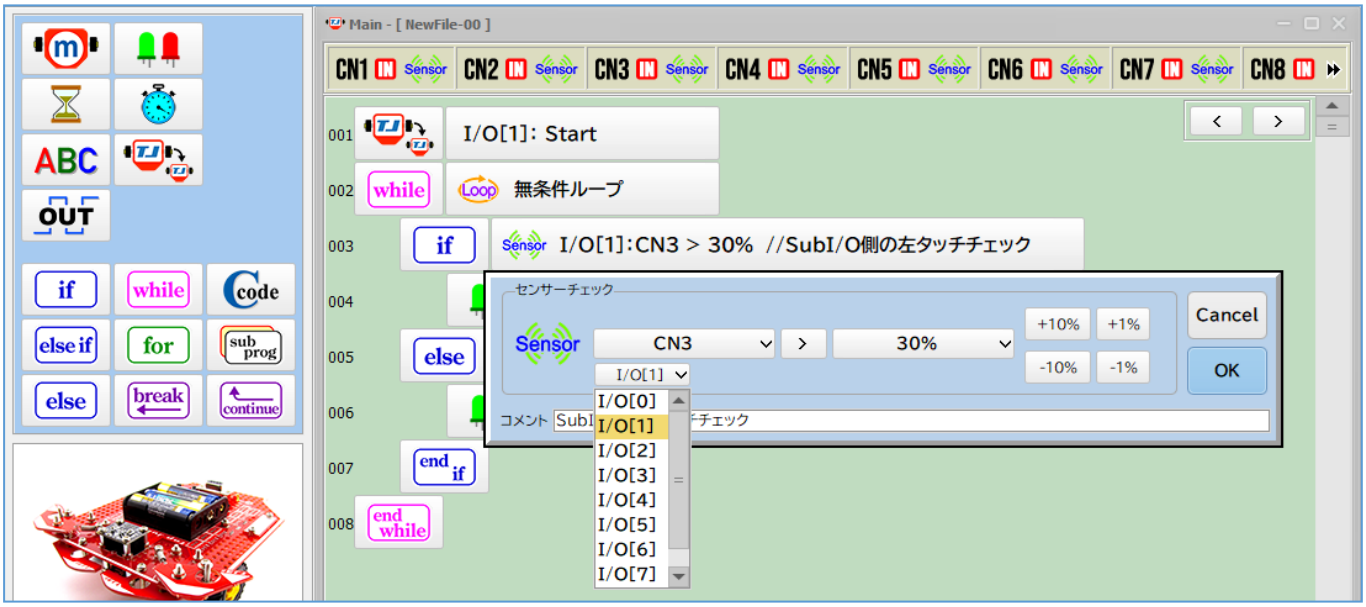
Setup 画面の ☑ Advanced Mode で ☑ I2C I/O ID Setup で Master を選択



プログラムボタンリストから SubI/O 制御ボタンの  を選択してプログラムの先頭行に置きます。
この  ボタンは Master 側のプログラムがスタートした時に SubI/O 側のプログラムも連動してスタートさせる命令です。



SubI/O 側のセンサーチェックを Master 側で行う場合は、該当するセンサーCN を選択してその下に表示されているプルダウンメニューをクリックして該当する SubI/O の ID を選択します。

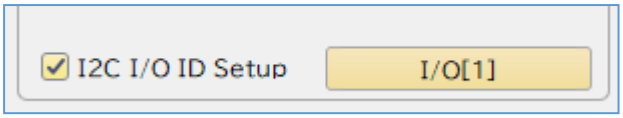


プログラム編集を終えたらビルドし、ダウンロードすれば Master 側の準備は完了です。

このプログラムは、SubI/O の左タッチで Master 側の LED が点灯するプログラム例です。

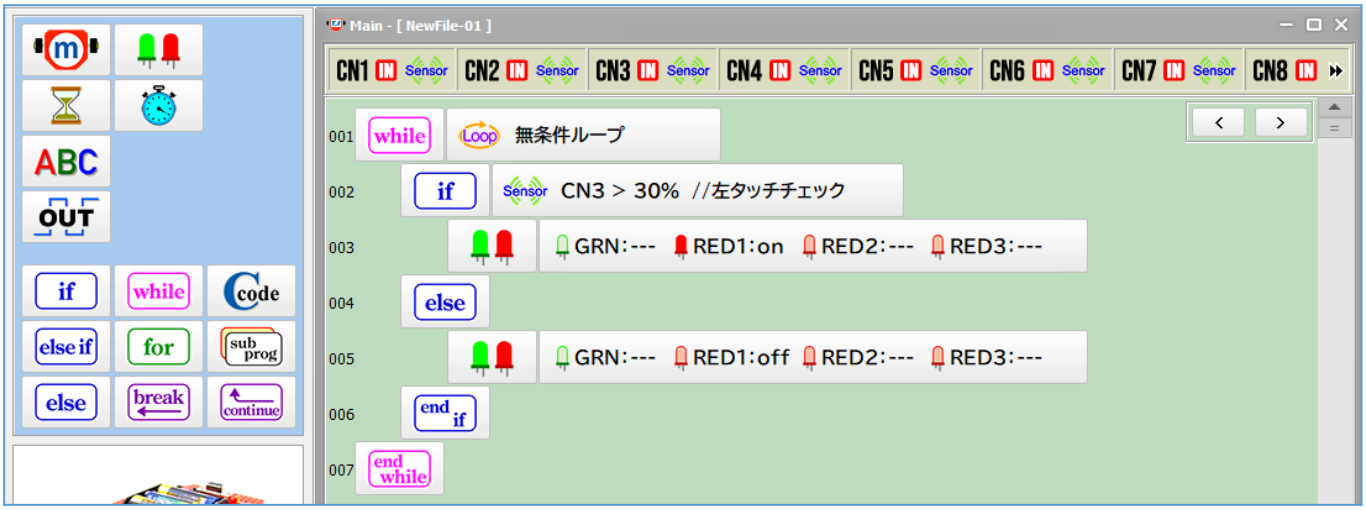
■SubI/O 側のプログラム編集

Setup 画面の ☑ Advanced Mode で ☑ I2C I/O ID Setup で I/O[1]を選択



Master 側のプログラムが単純に SubI/O 側のセンサー状態を得るだけであれば SubI/O 側のプログラムは空っぽの無条件ループだけのプログラムをダウンロードしておけば SubI/O がプログラム動作の待機状態（緑 LED の点滅）でもセンサー状態は Master 側に伝わる仕組みになっています。

下図の例では SubI/O 側のプログラムが動作するとタッチチェックで自身の LED を点灯制御するプログラムを示しています。



プログラム編集を終えたらビルドし、ダウンロードすれば SubI/O 側の準備は完了です。

Master 側のプログラムに SubI/O スタート命令がある場合はプログラムスタートで SubI/O のプログラムが自動でスタートします。この時 SubI/O 側の待機状態を示す緑 LED の点滅が消灯しプログラムがスタートしたことがわかります。SubI/O 側の左タッチをすると自身の LED も点灯しますが、Master 側の LED も点灯し SubI/O 側の情報が伝わっていることが確認できるはずです。

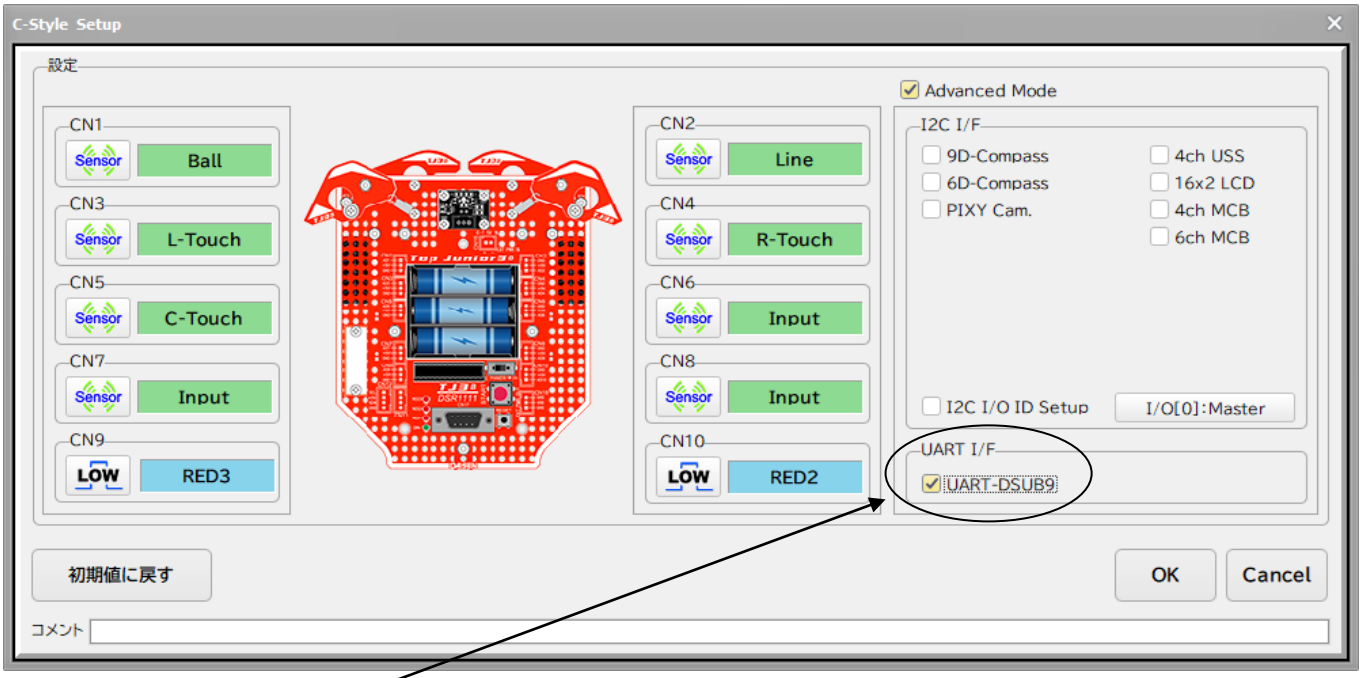
Master 側のプログラムに SubI/O スタート命令が無い場合ではプログラムスタートで SubI/O は待機中のままですが、タッチ操作を行うと Master 側の LED はそれに伴って動作します。

4-6. データロギング機能

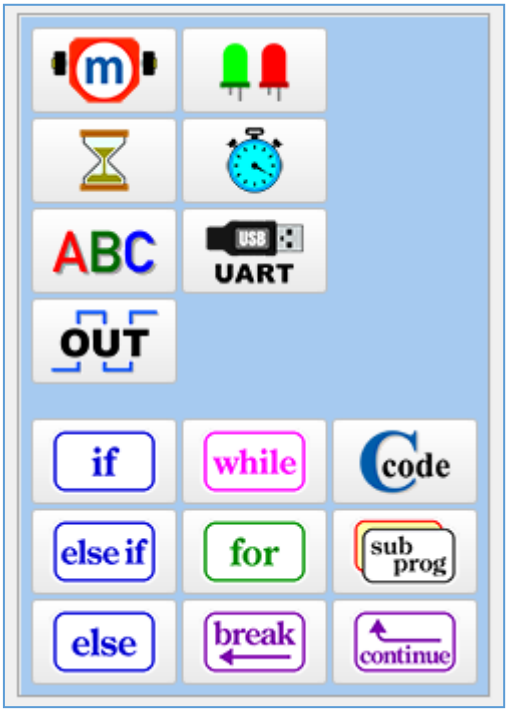
データロギングは、ロボットが停止中に行うセンサーモニターと違って C-Style プログラムが実行中に把握するセンサー情報や変数情報をダウンロード時に使用する通信ケーブルを通じて C-Style のデータロギング画面に表示及びファイルへの保存を行う為にある C-Style のプログラム機能です。

■データロギングの準備

入出力設定画面の☑Advanced Mode にチェックつけて拡張機能の設定画面を表示させます。



UART I/F の ☑ UART-DSUB9 にチェックを付けて「OK」ボタンクリックで画面を閉じます

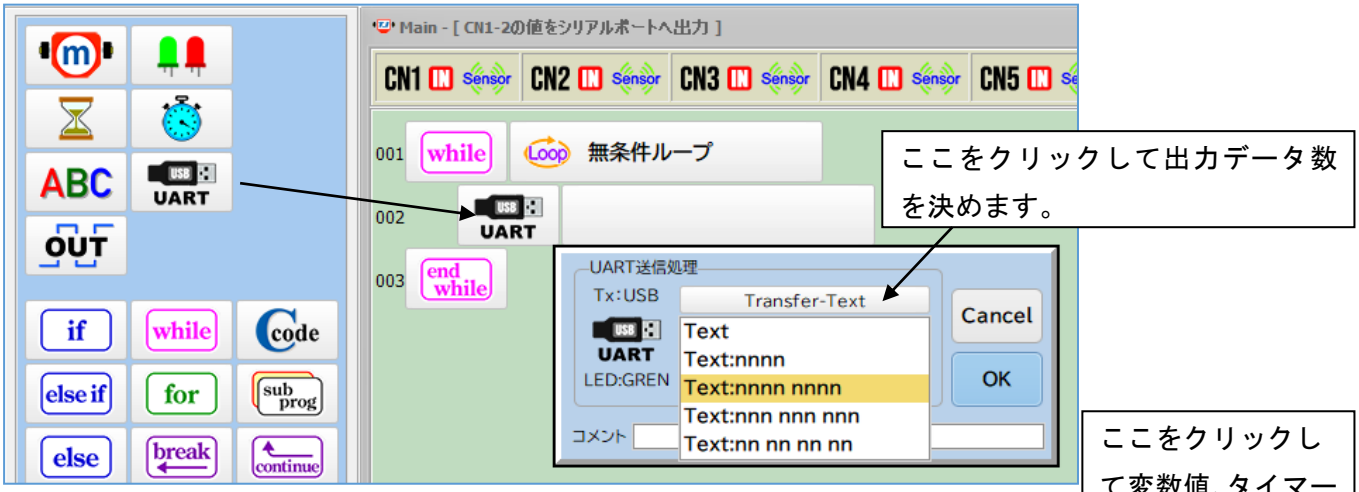


プログラムボタンリストに 

ボタンが追加されます。

■データロギングのプログラム作成

CN1 と CN2 のセンサー値をデータロギングするプログラム例

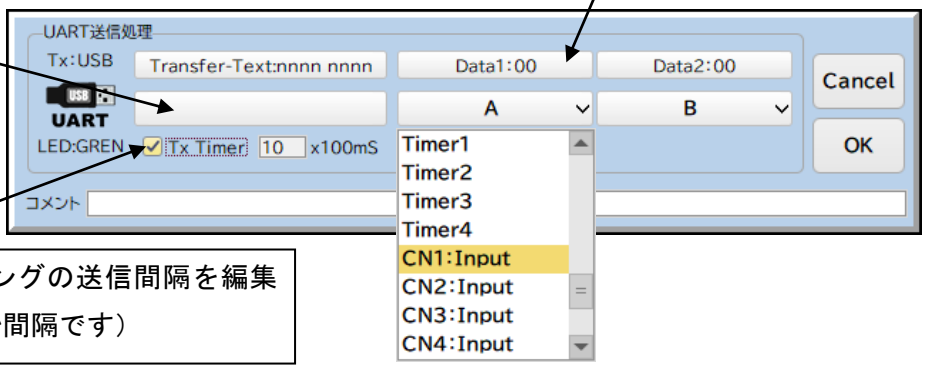


表示したいデータ数が CN1 と CN2 の 2 個の場合 “Text:nnnn nnnn” を選択します。

ここをクリックして変数値、タイマー値、センサー値の何れかを選択

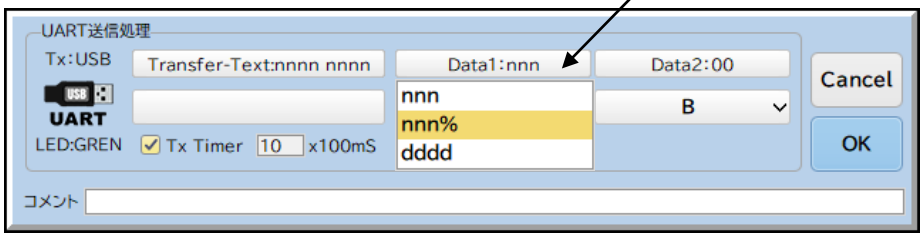
初期値は表示フォーム:“00” で変数 A と変数 B の値を出力する設定になります。

ここをクリックすると Text 文字列の編集が出来ます。（初期値は表示無し）

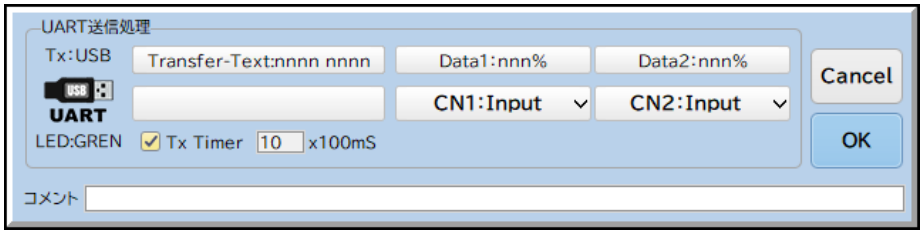


ここをクリックしてデータロギングの送信間隔を編集します。（初期値は 10x100mS=1 秒間隔です）

ここをクリックして表示フォームを選択



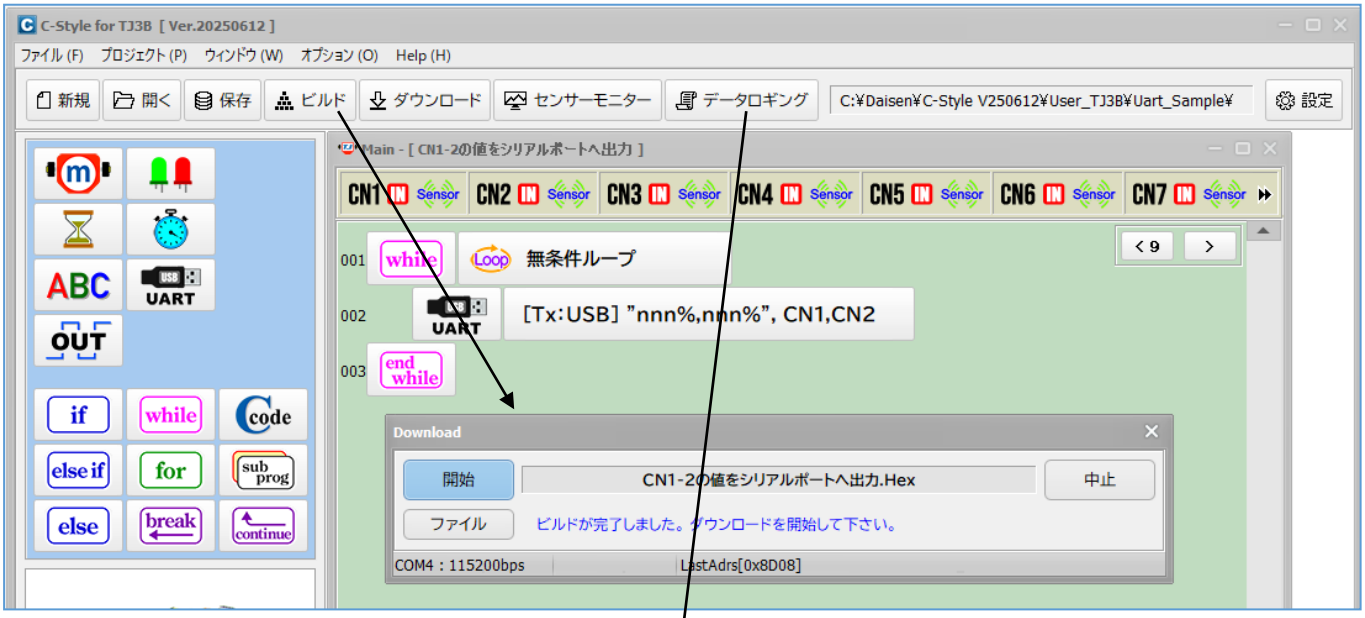
nnn は 0～100%の値で単位無し
nnn%は 0～100%の値で単位有りのフォーム
dddd は 0000～1023 の A/D 値のフォーム

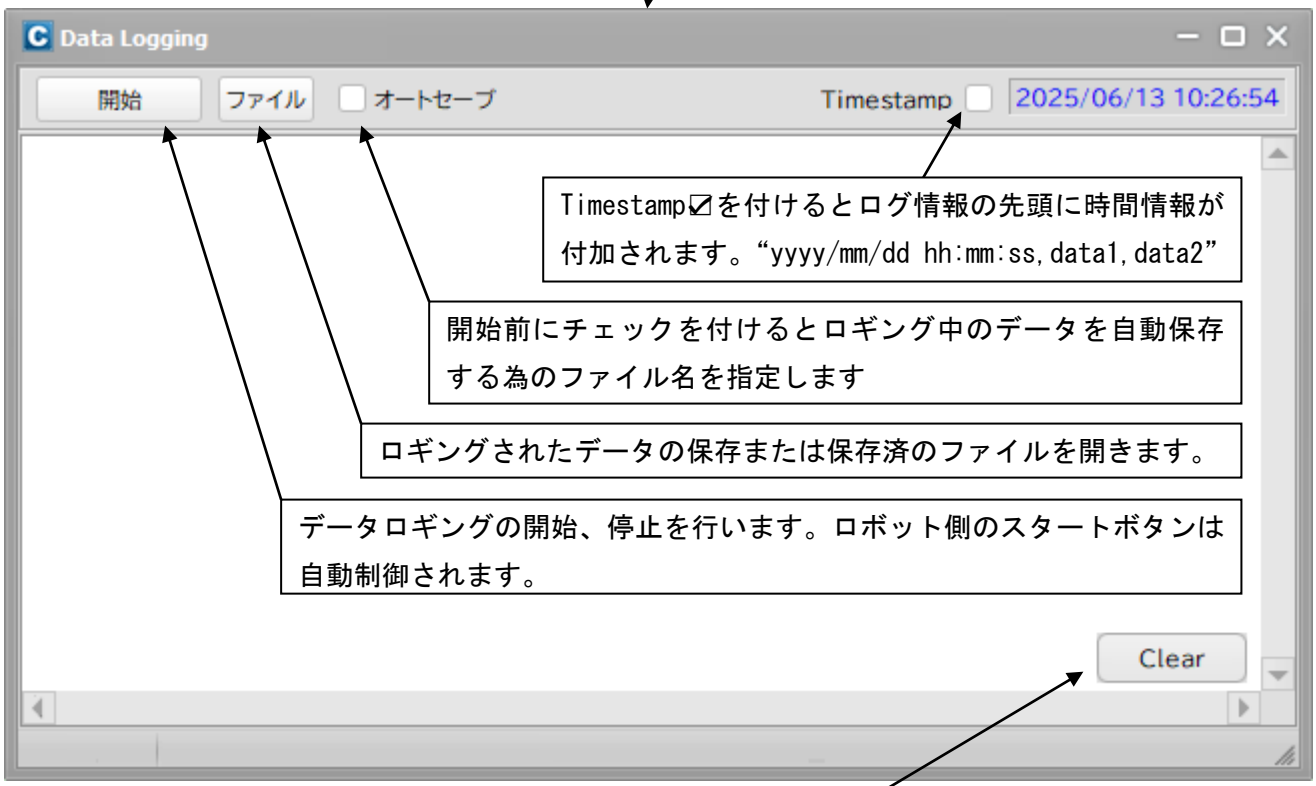


データ種別は CN1 と CN2 のセンサー値を　Data1,Data2 の出力フォームは “nnn%” の形式で設定し最後に「OK」ボタンクリックで決定します。

■データロギングの実行

データロギング用のプログラムをビルド・ダウンロード後通信ケーブルは外さないで、「データロギング」のボタンをクリックします。





Timestamp☑を付けるとログ情報の先頭に時間情報が付加されます。“yyyy/mm/dd hh:mm:ss,data1,data2”

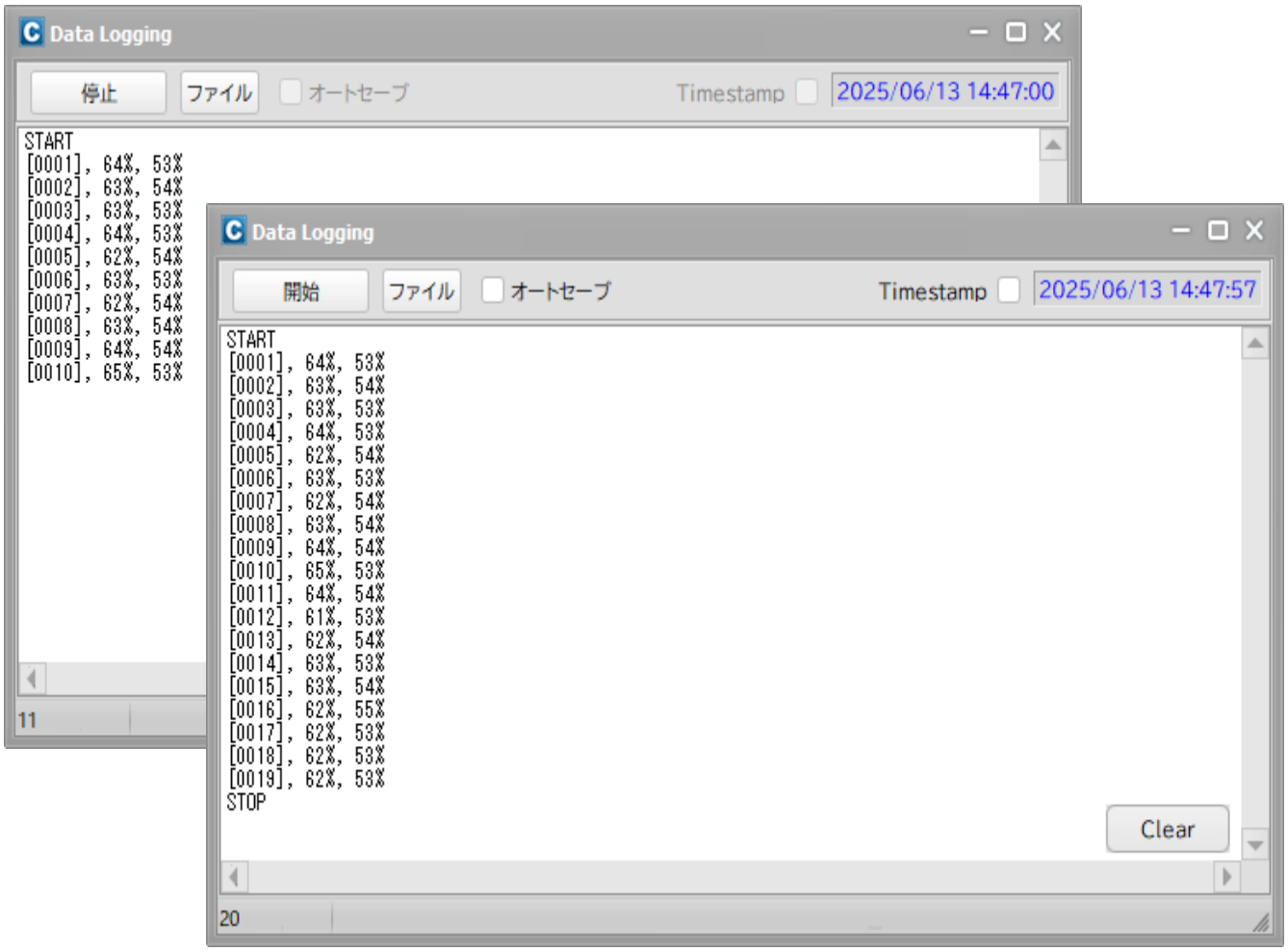
開始前にチェックを付けるとロギング中のデータを自動保存する為のファイル名を指定します

ロギングされたデータの保存または保存済のファイルを開きます。

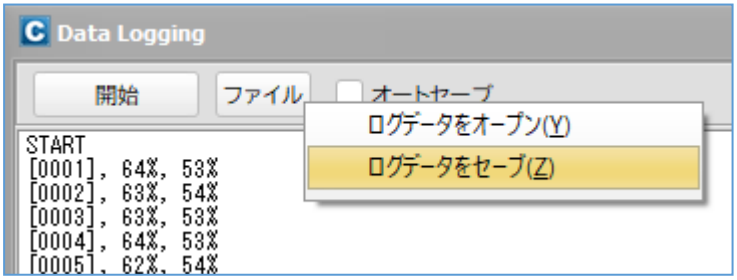
データロギングの開始、停止を行います。ロボット側のスタートボタンは自動制御されます。

「Clear」ボタンをクリックすると表示されているログデータが消去されます。

開始ボタンでデータロギングが開始されデータが画面に表示されます。
ロギング中の開始ボタンは停止ボタンに表示が変ります
停止ボタンでロギングが停止されます。（開始ボタンの表示もどります）



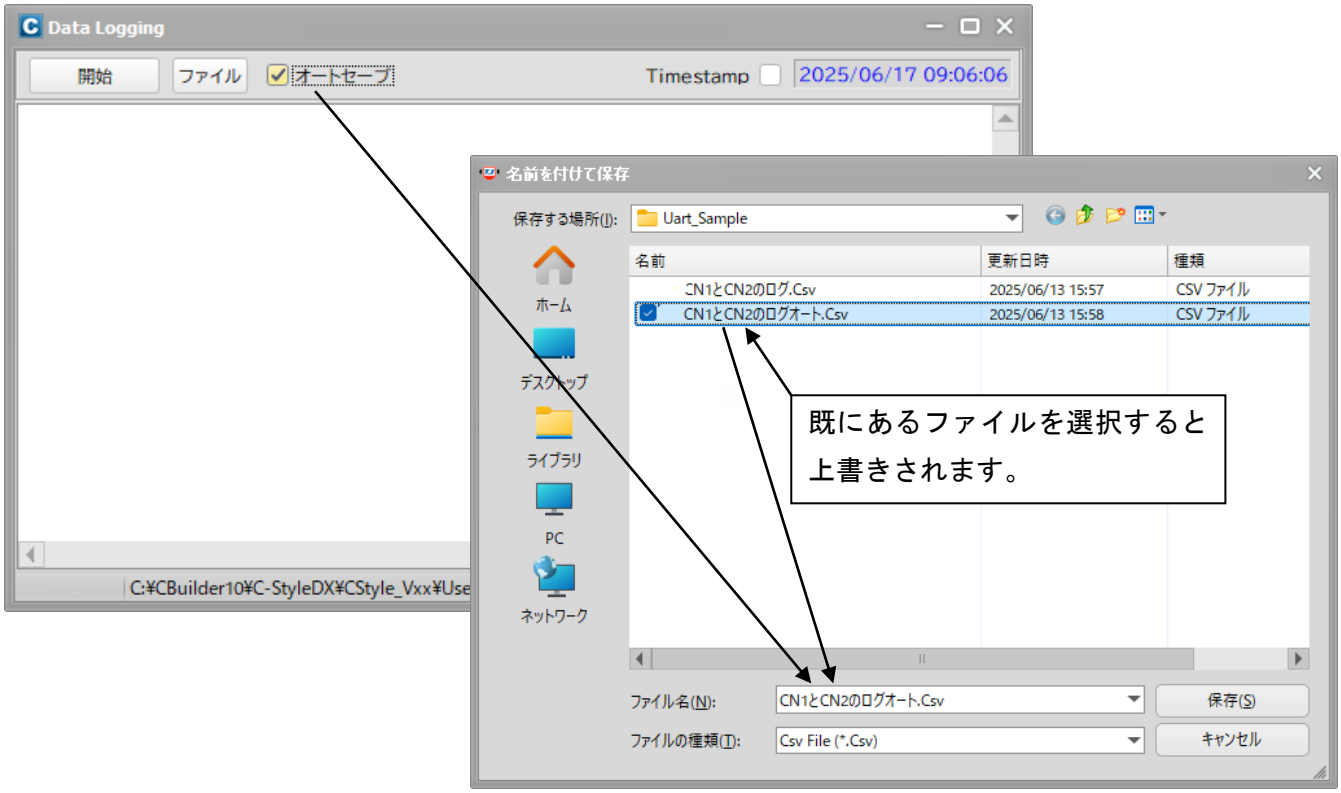
表示データを保存する場合は「ファイル」ボタンをクリックして「ログデータをセーブ」を選択して保存するファイル名を任意に作成して保存します。



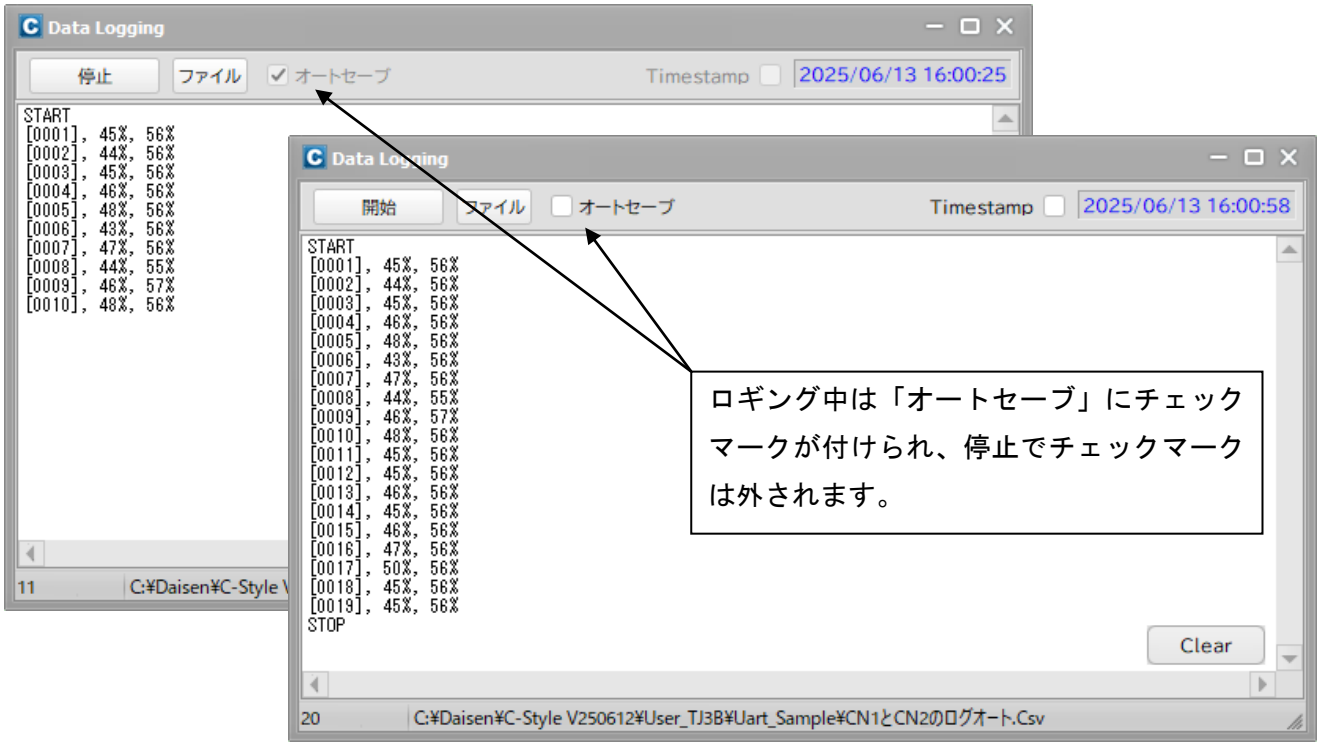
■データロギングの自動保存

ロギングデータをオートセーブ（自動保存）する場合は☑オートセーブにチェックを付けます。

名前を付けて保存のダイアログが表示されるので、自動保存するファイル名を事前に作成します。



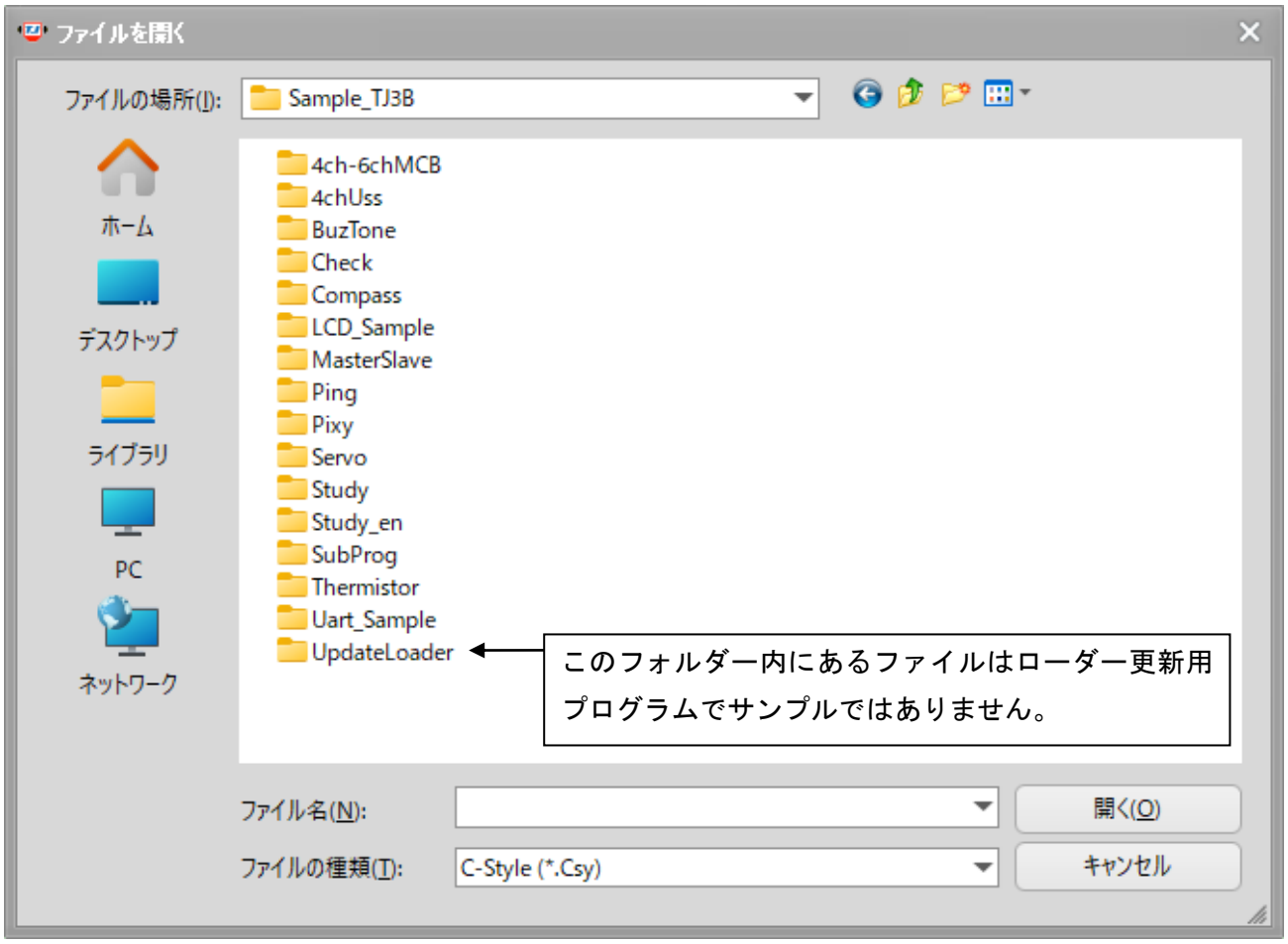
ロギングの「開始」から「停止」までのログデータを予め作成されたファイルに保存されます。



５．サンプルプログラム

5-1. サンプルプログラムフォルダー

C-Style をインストールしたフォルダー内に「User_TJ3B」というフォルダーが作成されています。
そのフォルダー内にある「Sample_TJ3B」フォルダーには各種サンプルプログラムが多数収納されています。
プログラム作成の参考にして下さい。

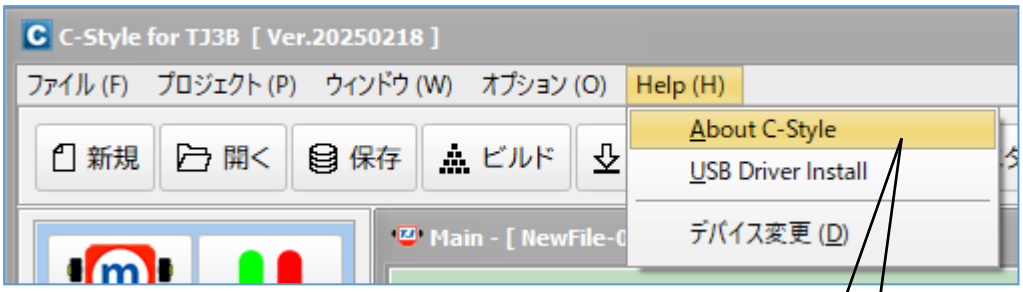


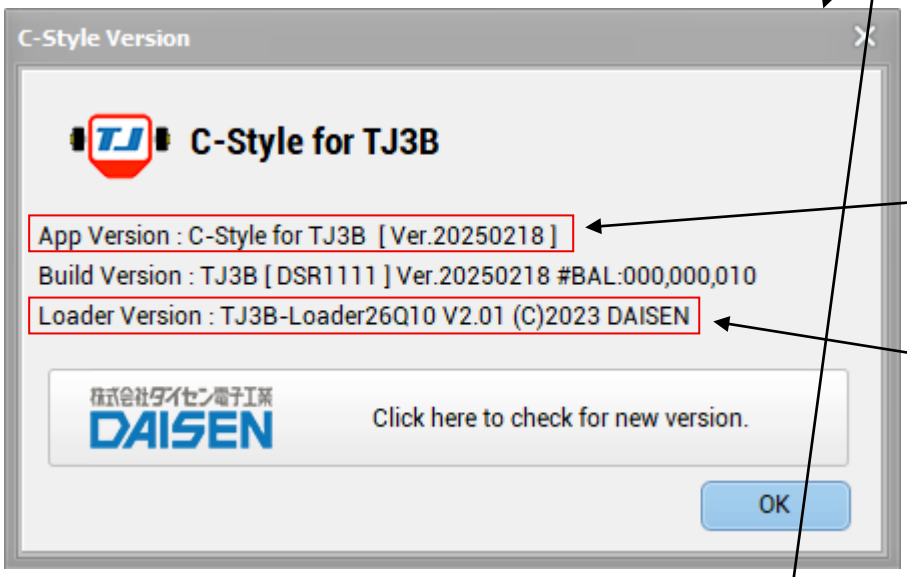
- 4ch-6chMCB ----- I2C 4ch・6ch モーターコントールボード使用サンプル
- 4chUss ---------- I2C 4chUss(Parallax 製 4ch 超音波距離センサーボード)使用サンプル
- BuzTone -------- メロディーブザー(DSR1801)使用サンプル
- Check -------- ロボット動作チェック用プログラム（ロボット内蔵チェックプログラムと同等）
- Compass -------- I2C 9D-Compass 使用サンプル
- LCD_Sample ------ I2C 16x2 LCD 使用サンプル
- MasterSlave ----- I2C 複数台の TJ3B を接続するサンプル
- Ping ---------- Parallax 製 超音波距離センサーを CN7～CN10 に直接接続して使用するサンプル
- Pixy ---------- I2C PixyCam.イメージセンサー使用サンプル
- Servo ---------- 低トルクサーボモーターを CN6～CN10 に直接接続して使用するサンプル
- Study / Study_en --- 入門編サンプル、_en:英語版
- SubProg ---------- サブプログラムサンプル
- Thermistor ------ サーミスター温度センサーを CN1～CN10 に直接接続して使用するサンプル
- Uart_Sample ----- ダウンロードケーブルを使用して行うデータロギングサンプル

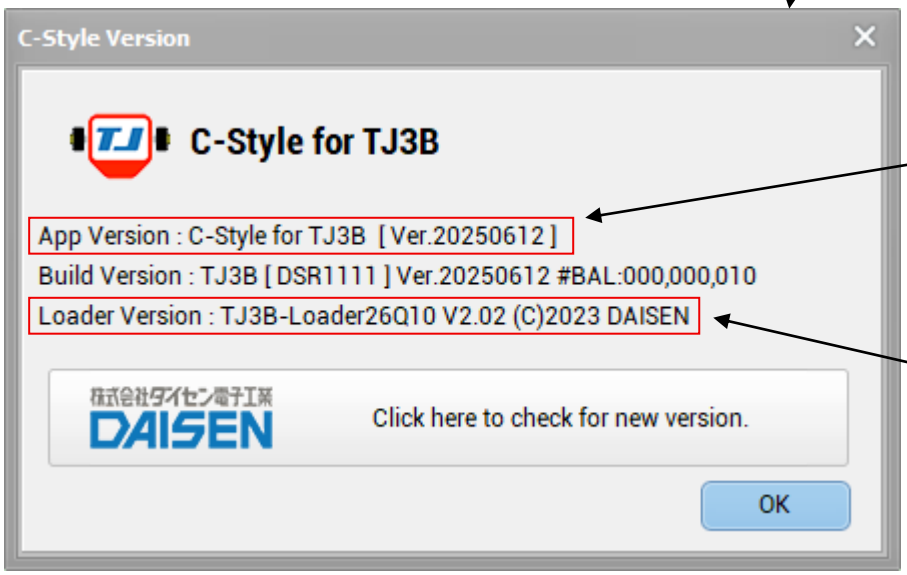
６． ロボットのダウンローダーを更新

ローダープログラムは、C-Style をビルドしてロボットへダウンロードする際に実行される特別なプログラムでロボット側の Flash Rom に特別な装置で書き込まれています。そのプログラムを特別な装置無しで書換えるプログラムを UpdateLoader と呼んでいます。

6-1. Loader バージョンの確認







① Help から About C-Style を選択して C-Style のバージョンを表示させます。

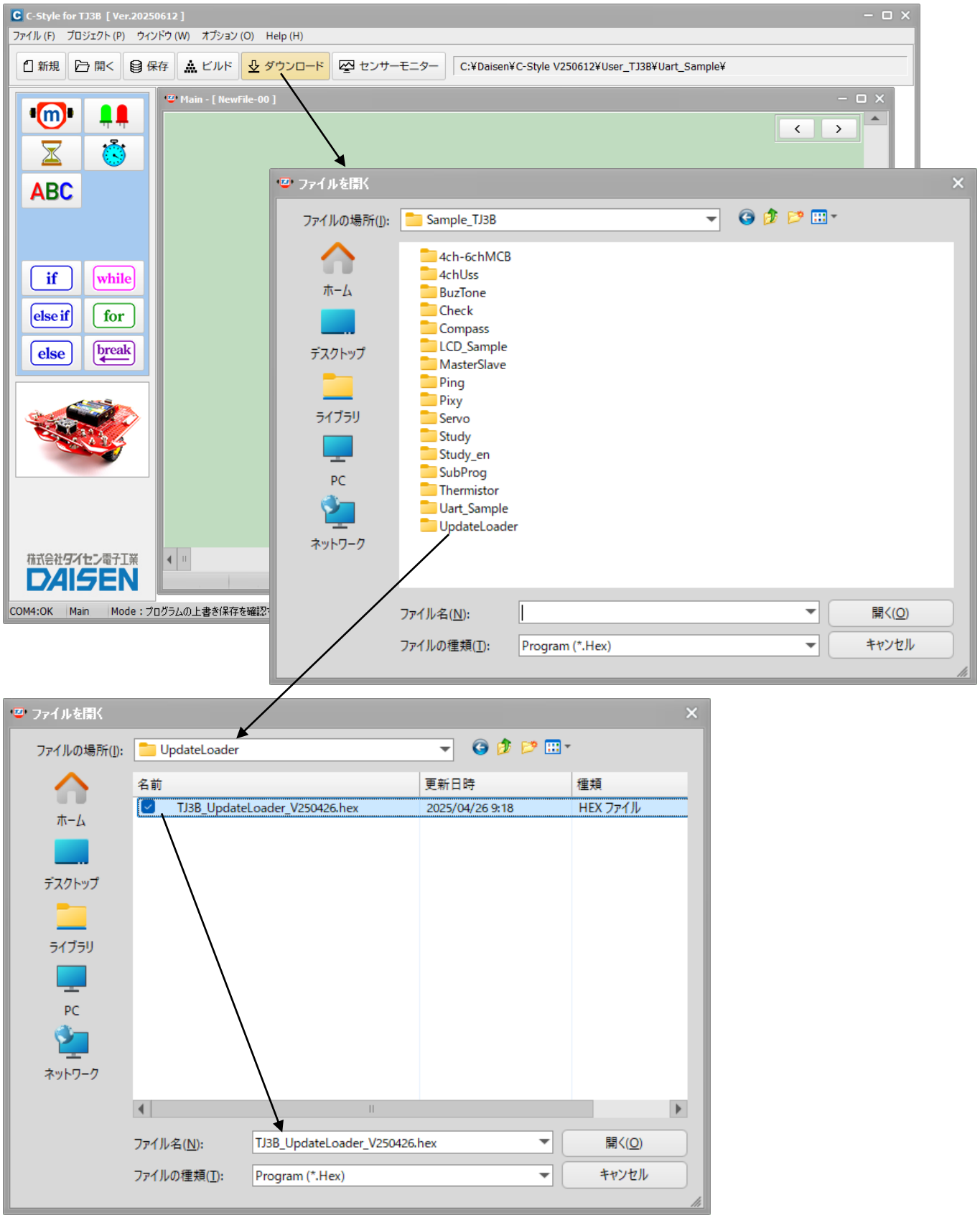
② App Version が Ver.20250218 以前の場合は C-Style を更新して下さい。

③ TJ3B-Loader26Q10 V2.00 または V2.01 の場合は、UpdateLoader を実行して更新することを推奨します。
※実行中にロボットがリセットして待機状態になるのを防ぐ改善がなされています。

④ UpdateLoader の最新版は App Version が Ver.20250621 の C-Style に実装されていますので UpdateLoader を実行する前に C-Style を更新して下さい。

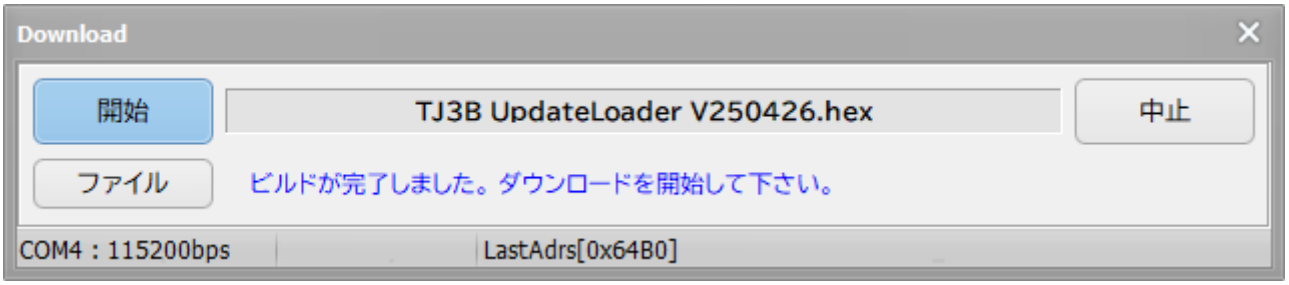
⑤ TJ3B-Loader が V2.02 であれば最新版ですので更新する必要はありません。C

6-2. UpdateLoader のファイルを選択



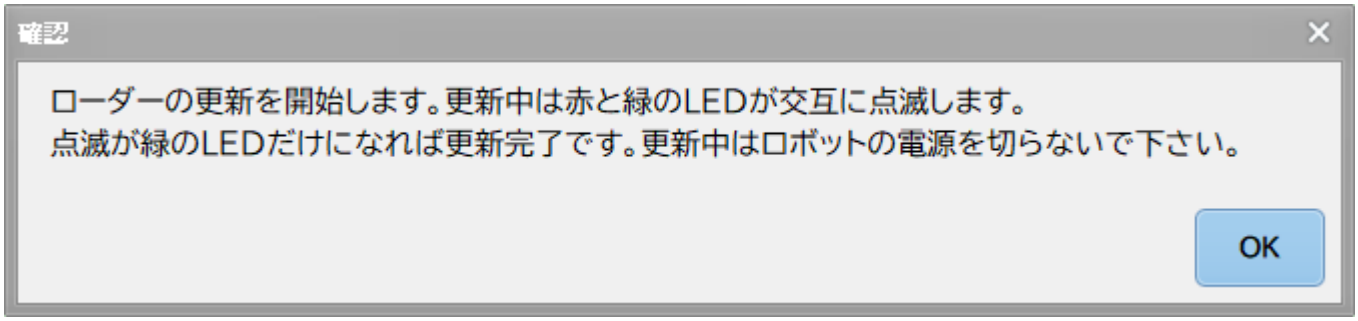
“TJ3B_UpdateLoader_V250426.hex” を選択して「開く」ボタンをクリックします。

6-3. UpdateLoader の実行



“TJ3B_UpdateLoader_V250426.hex” は TJ3B-Loader26Q10 V2.02 用の更新ファイルです。
通常のダウンロード同様に「開始」をクリックしてダウンロードを開始して下さい。

※ご注意
通信ケーブルは外さないで下さい。
ダウンロード完了のダイアログが通常と異なることに注意して下さい。
ダウンロードしたローダー更新プログラムを実行する必要があります。
ダウンロード完了ダイアログの「OK」ボタンをクリックしますと更新が開始されます。
その間ロボットの電源は切らないで下さい。



更新完了後は緑色 LED だけ高速点滅していますので、そのまま任意のプログラムをビルドしてダウンロードして下さい。その後センサーモニターを実行して正常に動作していれば更新成功です。
Help の About C-Style でローダーのバージョンを確認して下さい。

ローダーバージョンが V2.02 と表示していれば更新成功です。

⚠注意

本製品は一般の民生・産業用として使用されることを前提に設計されています。
人命や危害に直接的、間接的にかかわるシステムや医療機器など、高い安全性が必要とされる用途にはお使いにならないでください。

本製品の故障・誤動作・不具合によりシステムに発生した付随的障害および、
本製品を用いたことによって生じた損害に対し、当社は一切責任を負いません。
あらかじめご了承ください。

株式会社ダイセン電子工業
DAISEN
〒556-0005 大阪市浪速区日本橋 4-9-24
TEL：06-6631-5553 / FAX：06-6631-6886
URL：https://www.daisendenshi.com
e-mail：ddk@daisendenshi.com