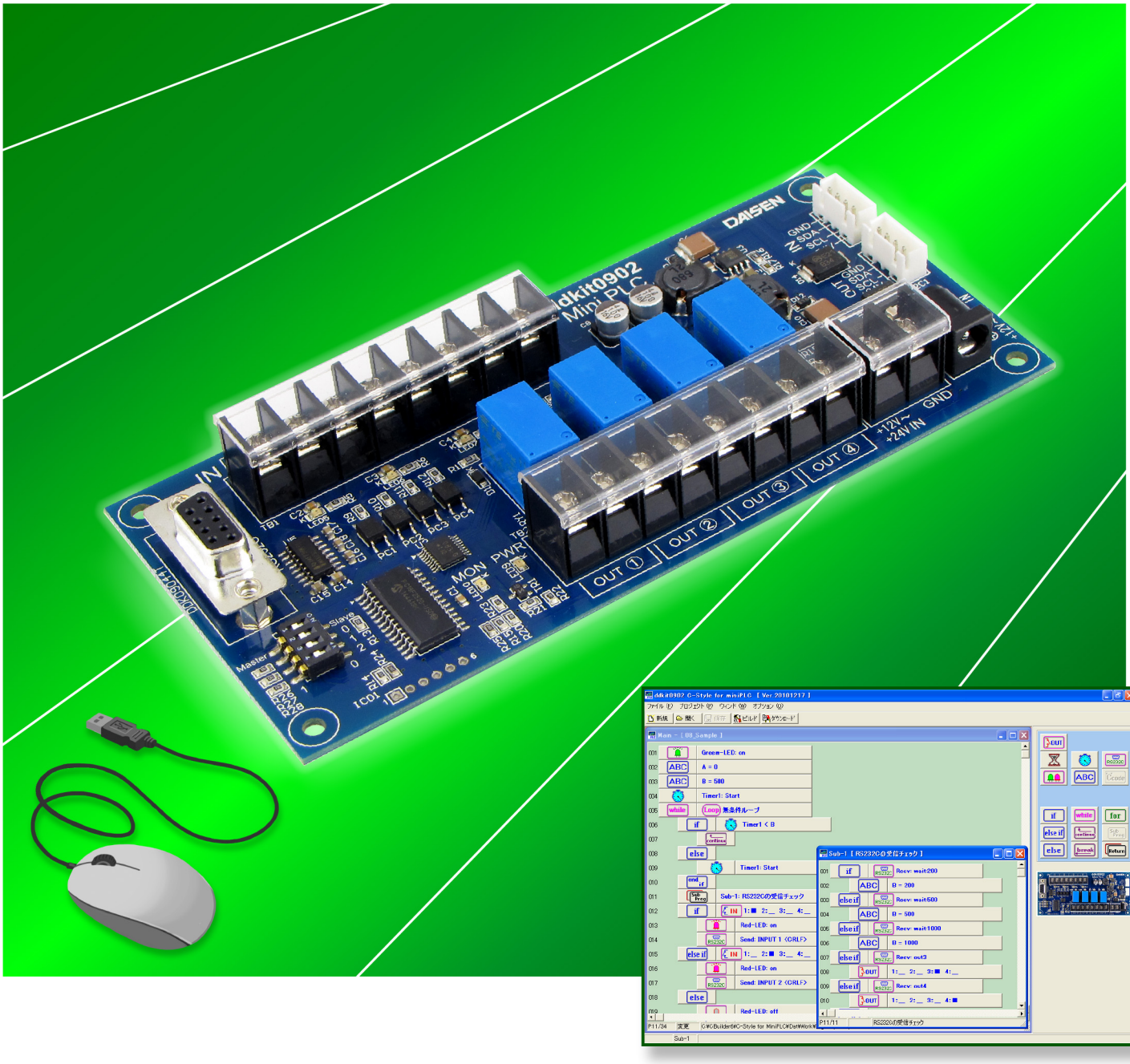
Mini PLC

C-Style 操作編



DAISEN

株式会社ダイセン電子工業

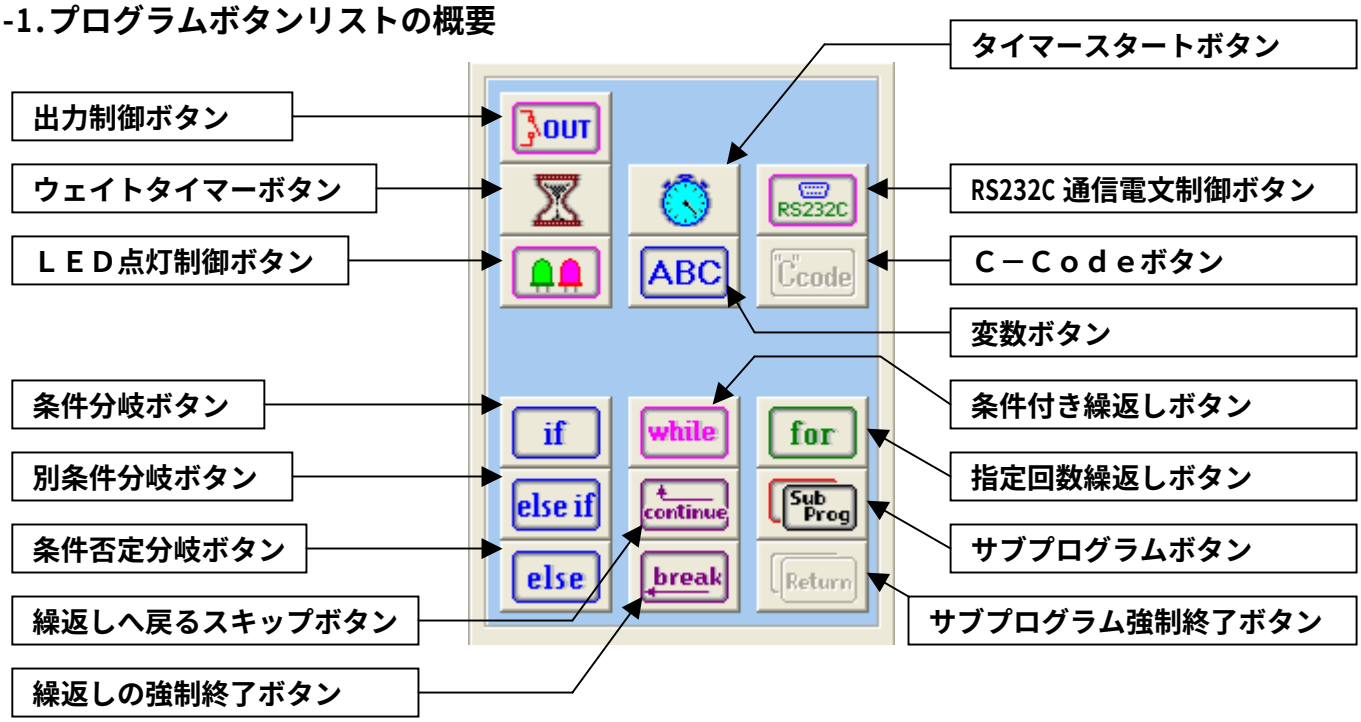
Rev.130912

目　次

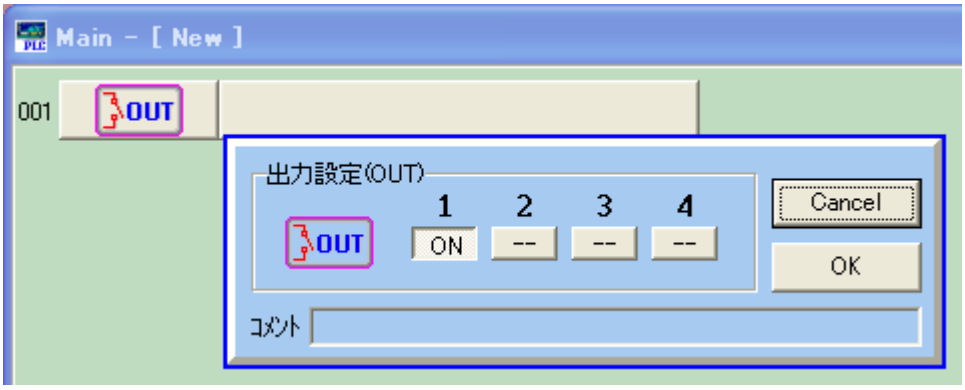
C-Style 操作ガイド（本書）

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の概要

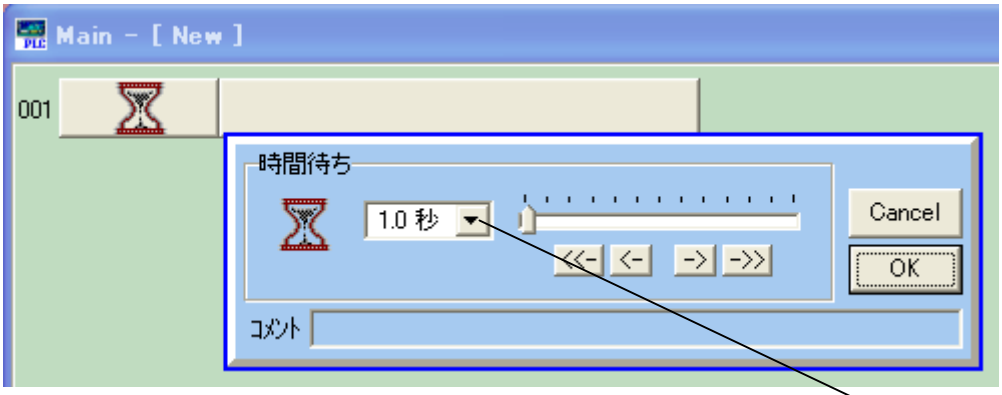


1-2.出力制御ボタン OUT

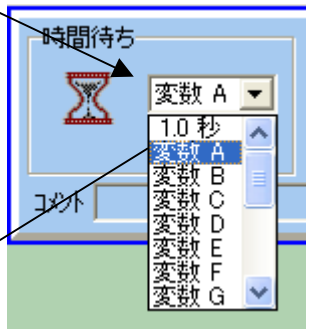


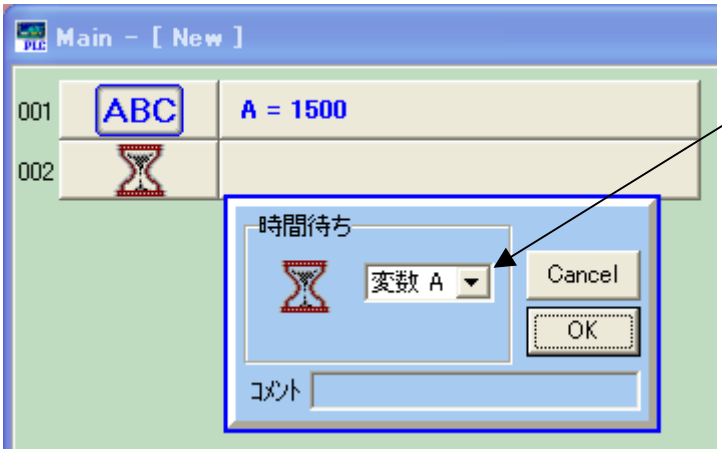
ＯＵＴ１～ＯＵＴ４の出力設定を行います。
該当する出力番号下のボタンをクリックしますと「ON」「OFF」「--」に順に表示が変わります。
「--」は出力状態を変更しない設定となります。
上の例ですと、ＯＵＴ１だけが「ON」で他は現在の状態を保持する設定となります。

1-3.ウェイトタイマーボタン



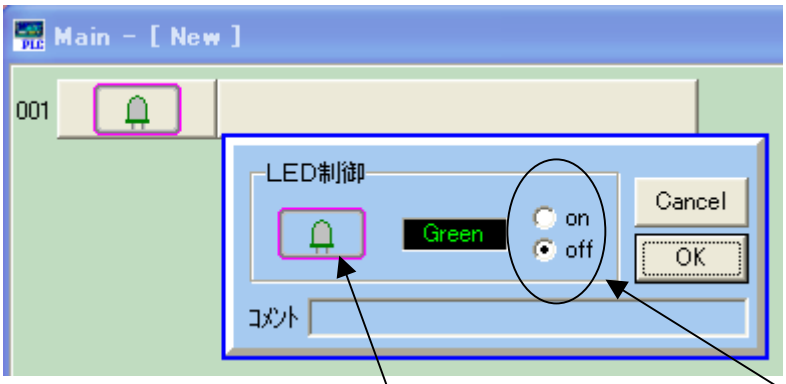
0.1 秒から 60.0 秒までの時間待ちを設定します。
待ち時間を変数に設定することも出来ます。
変数へ待ち時間を設定する場合は、1000 倍したミリ秒の単位で
設定します。
例えば、1.5 秒の待ち時間を変数へ設定する場合は、下図のよにします。





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出来ません。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。

1-4. ＬＥＤ制御ボタン 

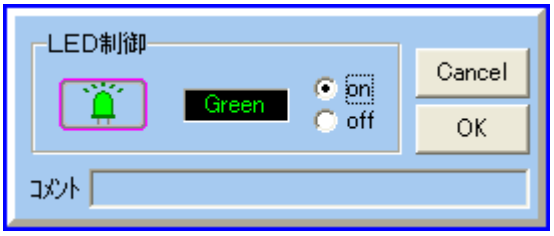


緑ＬＥＤと赤ＬＥＤの点灯設定を行います。

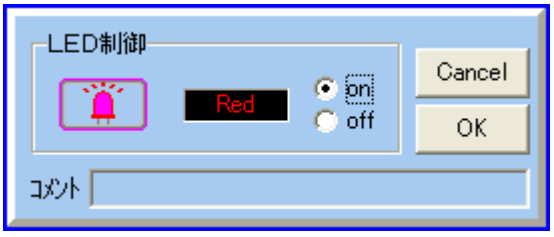
ここをクリックして赤と緑のＬＥＤを選択します。

ここをクリックして点灯、消灯を選択します。

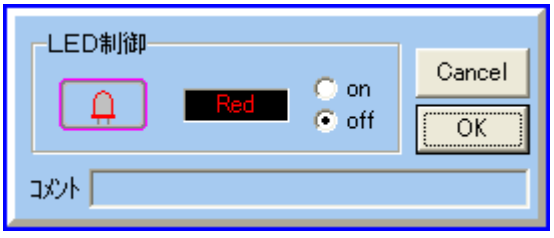
緑ＬＥＤの点灯



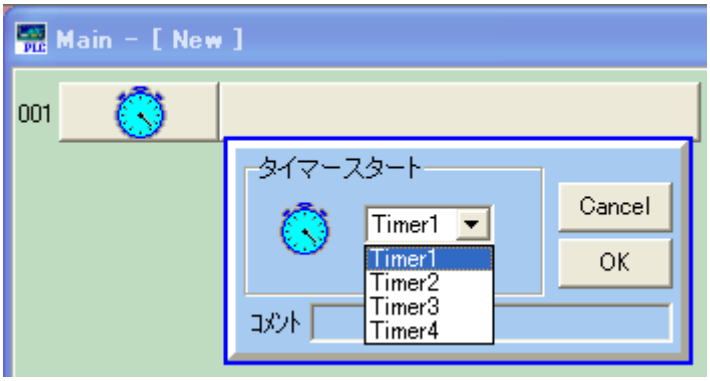
赤ＬＥＤの点灯



赤ＬＥＤの消灯



1-5.タイマースタートボタン

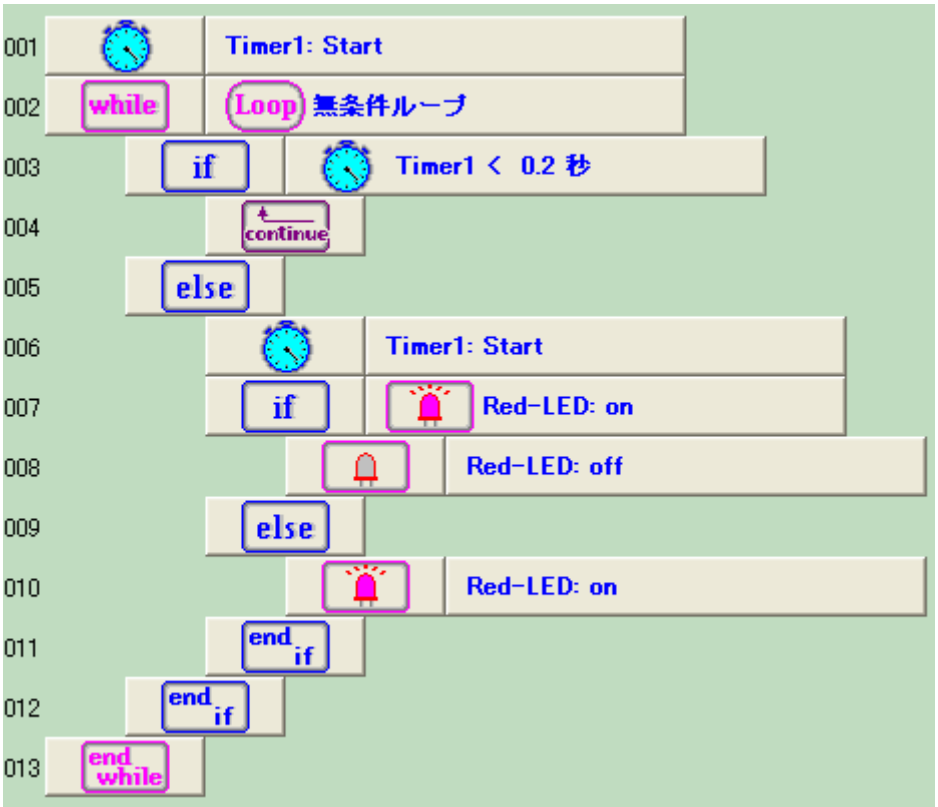


タイマースタートを実行するボタンです。
４個のタイマーを選択して使用します。

タイマーの経過は、条件判定のタイムチェックで判定します。

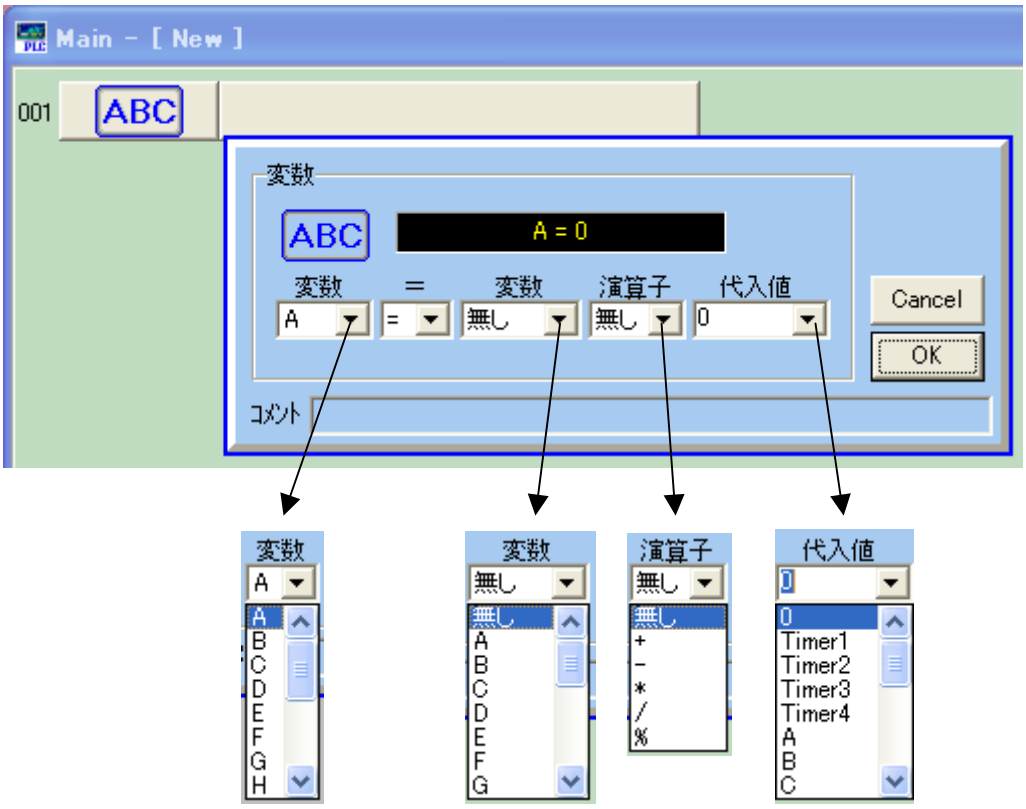
ストップウォッチのスタートボタンを押す感じと同じですが、停止はなく、条件判定で時間の経過を判定します。

タイマーを使ったプログラム例



この例は、０．２秒間隔で赤ＬＥＤの点滅をするプログラムです。
ウェイトタイマー と違って時間の経過中も他の処理が行えます。

1-6.変数ボタン ABC

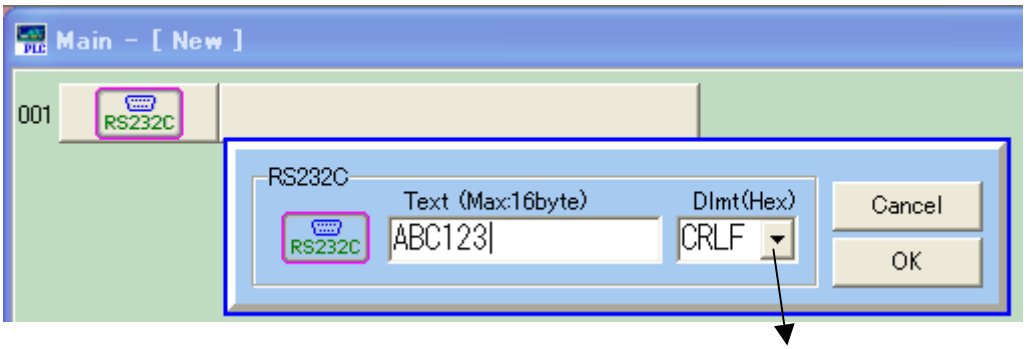


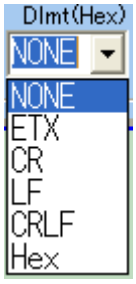
変数ボタンは、Ａ～Ｐの１６個の変数が使用出来ます。
演算子と＋、－、×、÷、余り計算などが行えます。
またタイマーの値や他の変数も代入することが出来ます。

◇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
“A = Timer1	変数 A にタイマー１の値を代入（変数 A は 0～65535 ﾐﾘ秒の値になる）
“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）

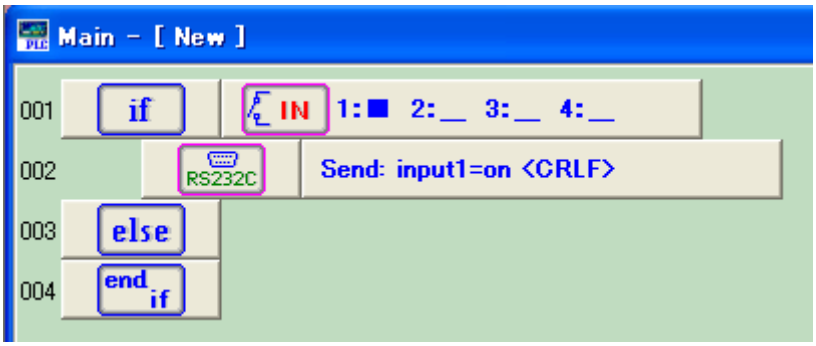
1-7.RS232C ボタン 





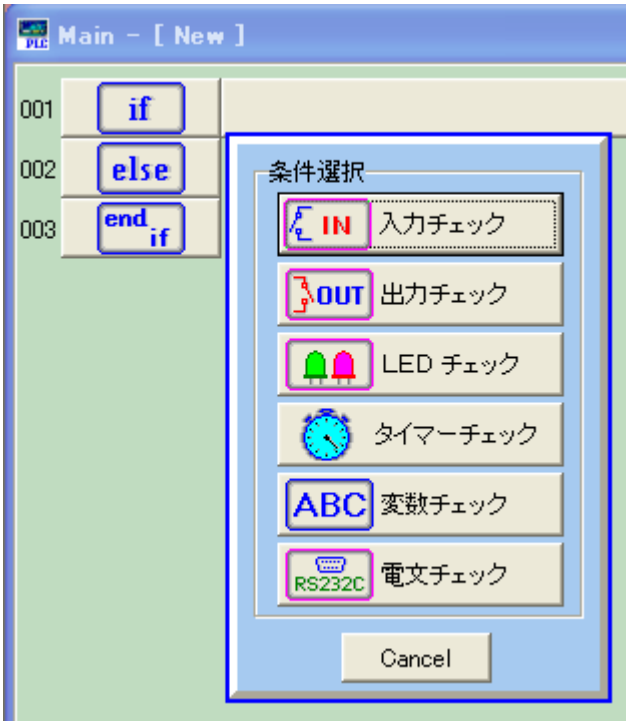
このボタンは、RS232C の通信 I/F に対して
１６文字までのテキストを送信することが
出来ます
送信デリミタとして以下のものが選択出来ます。
「NONE」：無し、「ETX」：0x03、「CR」：0x0D、
「LF」：0x0A、「CRLF」：0x0D＋0x0A の選択と
「Hex」：0x00～0xFF の値を直接入力出来ます。

ＲＳ２３２Ｃプログラム例



入力１がＯＮの時、“ input1=on ” のテキストを<CRLF>のデリミタを付けて送信する。

1-8.条件分岐



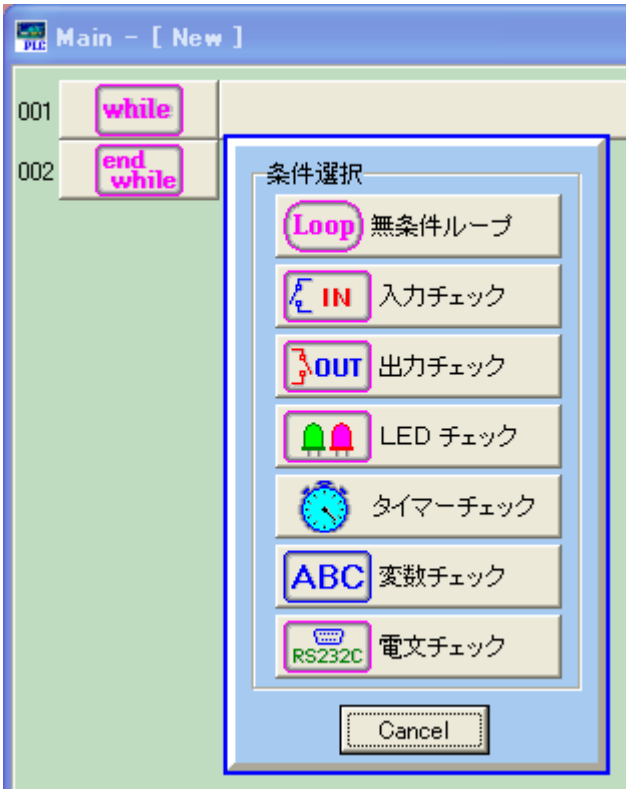
if イフと呼びます。(条件成立で次の行からプログラムを実行します。)
条件成立で、この間に置かれたプログラムボタンが実行されます
else if エルス イフと読みます。(条件成立で次の行からプログラムを実行します。)
条件成立で、この間に置かれたプログラムボタンが実行されます。
else エルスと呼びます。(条件不正立の場合に次の行からプログラムを実行します。)
どの条件も成立しなかった時、この間に置かれたプログラムボタンが実行されます。
end if エンド イフと呼びます。

「if」と「else」「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、入出力チェック、ＬＥＤチェック、タイムチェック、変数チェック、ＲＳ２３２Ｃの電文チェックの判定条件を選択することが出来ます。
「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。

1-9.条件付き繰返し



while ホワイルと呼びます。
繰返しの条件が成立している間、この間に置かれたプログラムボタンの処理を繰返し実行されます。
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）

繰返しの中でだけ使えるボタンに continue break があります。

continue はコンティニュウと呼びます。 このボタンに出会うと、一番近い while ボタンに戻ります。（「end while」ボタンまでのプログラムボタンは実行されません）

break はブレイクと呼びます。 このボタンに出会うと、繰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ず end while ボタンの次のボタンから実行します。）

1-10.回数指定の繰返し

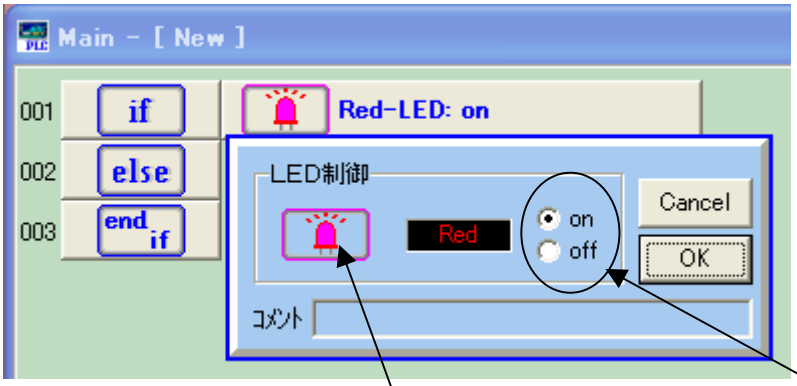
for　フォアと呼びます。
指定回数、この間に置かれたプログラムボタンを実行します。
end for　エンド フォアと呼びます。

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）

continue　コンティニュウボタンが使えます。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）

break　ブレイクボタンが使えます。　このボタンに出会うと、繰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）

1-11.ＬＥＤチェック

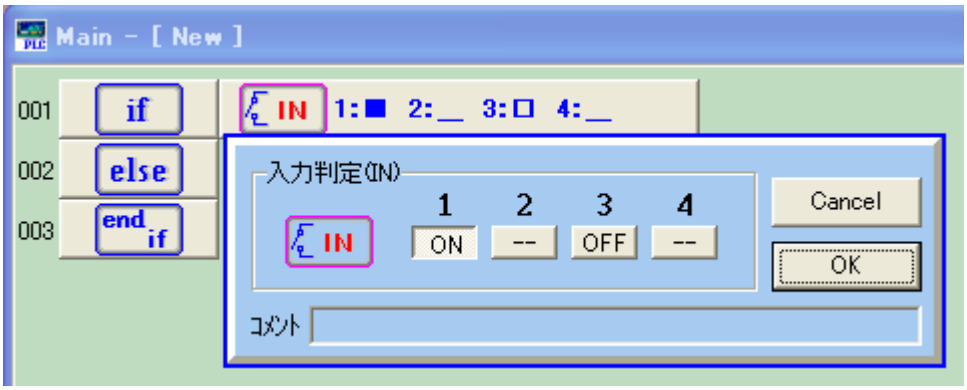


ＬＥＤの点灯状況を判定して条件分岐します。

ここをクリックして赤と緑のＬＥＤを選択します。

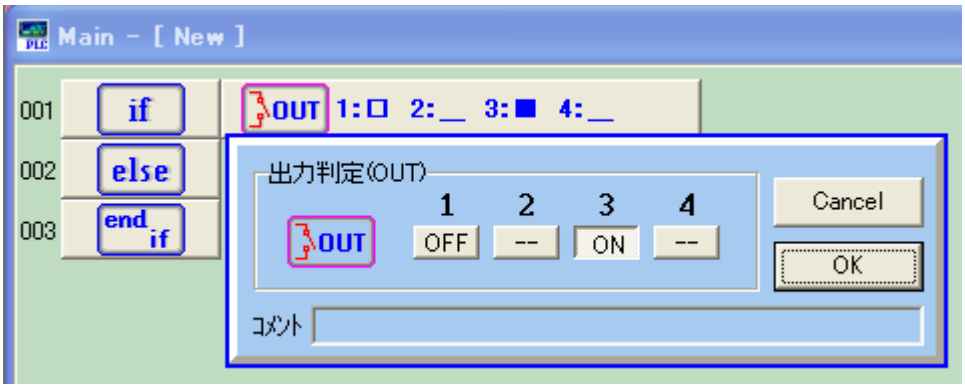
ここをクリックして点灯、消灯の判定条件を選択します。

1-12.入力チェック



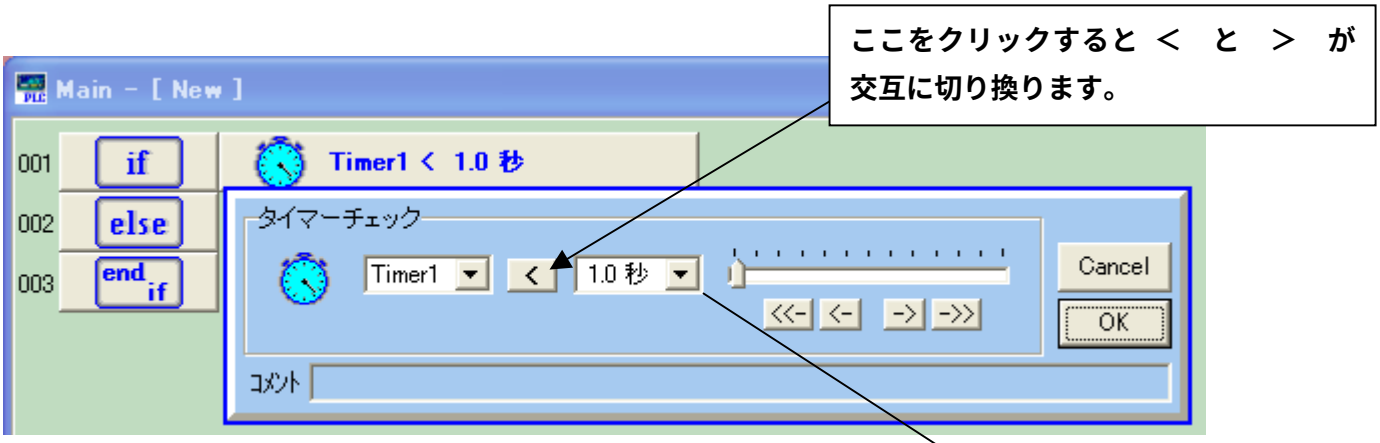
ＩＮ１～ＩＮ４の入力状態を判定して条件分岐します。
各入力番号下のボタンをクリックすると、「ON」「OFF」「--」の順に表示が変わり、判定条件を設定します。
「ON」 は入力有りを判定します。
「OFF」は入力無しを判定します。
「--」 は入力状態をマスクします（判定しません）
例では、ＩＮ１＝ＯＮ で ＩＮ３＝ＯＦＦ の場合に条件成立となります。（入力２，４は判定しない）

1-13.出力チェック

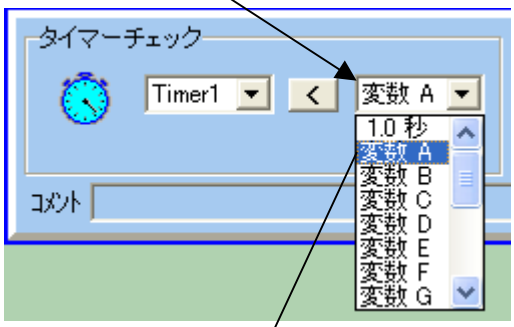


ＯＵＴ１～ＯＵＴ４の出力状態を判定して条件分岐します。
各出力番号下のボタンをクリックすると、「ON」「OFF」「--」の順に表示が変わり、判定条件を設定します。
「ON」 は出力有りを判定します。
「OFF」は出力無しを判定します。
「--」 は出力状態をマスクします（判定しません）
例では、ＯＵＴ１＝ＯＦＦ で ＯＵＴ３＝ＯＮ の場合に条件成立となります。（出力２，４は判定しない）

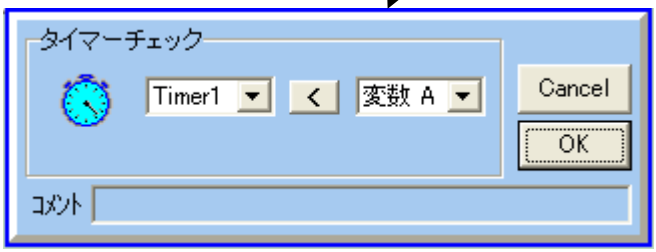
1-14.タイマーチェック



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出来ます。
設定値のかわりに、変数（Ａ～Ｐ）との比較判定も出来ます。

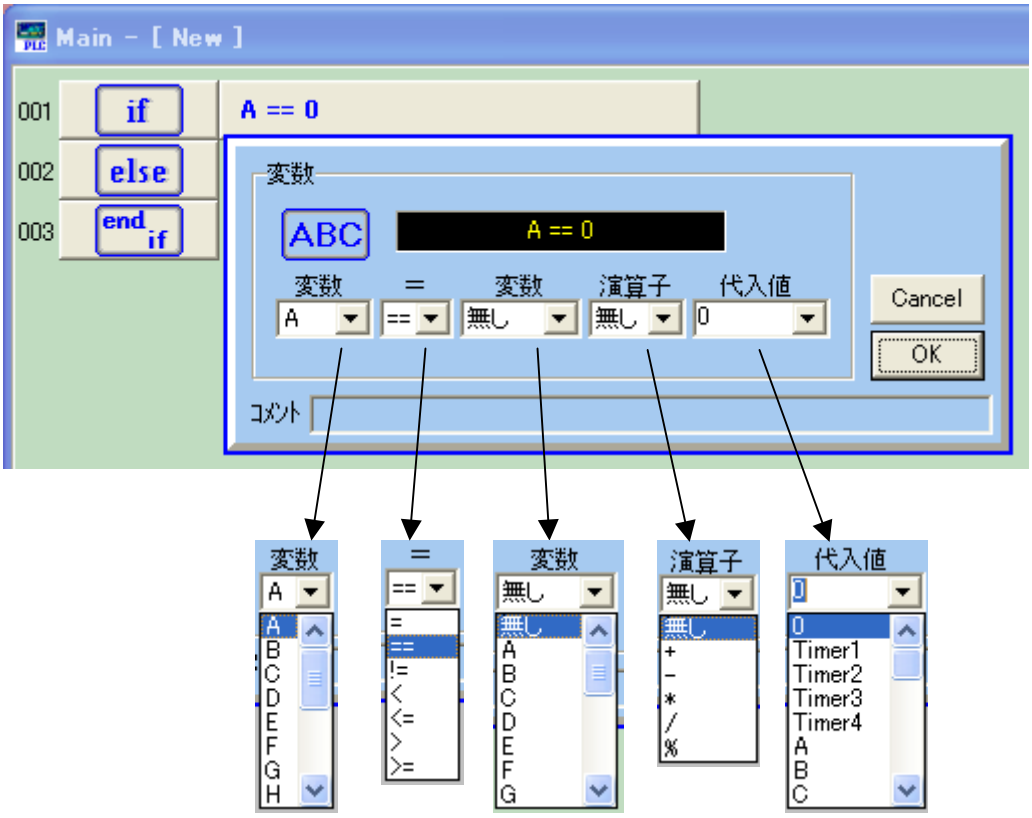


ボタン名称の例
“Timer1 ＜ 0.1秒以下”
“Timer2 ＞ 1.5秒以上”
“Timer1 ＞ 変数Ａ 以上”
“Timer2 ＜ 変数Ａ 以下”



◇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

1-15.変数チェック

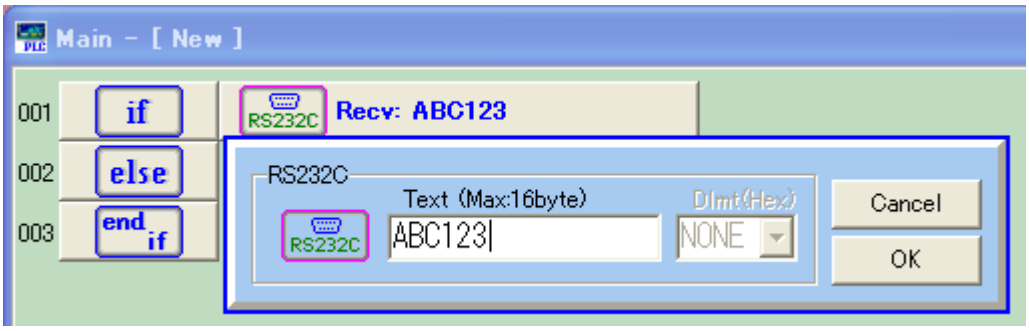


変数は、Ａ～Ｐの１６個が使用出来ます。
演算子と＋、－、×、÷、余り計算などが行えます。
また、タイマーの値やCN1からCN10に接続されたセンサーの値も変数へ代入したり比較することが出来ます。

ボタン名称の例

“A == 0”	変数Aが0の場合
“A ＜ 5”	変数Aが5以下の場合
“A ＞ B”	変数Aが変数Bより大きき場合
“A ＜ B － C”	変数Aより変数Bから変数C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 ＜ Timer3”	変数Aよりタイマー３の値の方が大きい場合

1-16. ＲＳ２３２Ｃ電文チェック



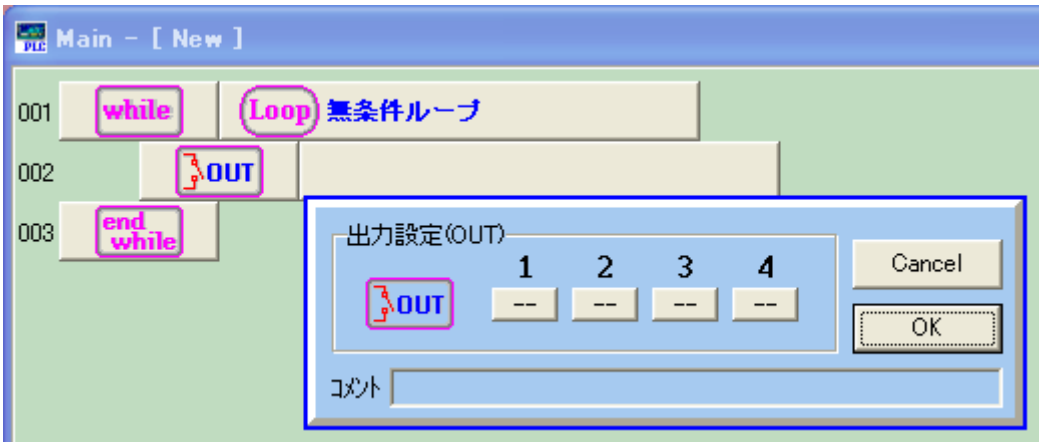
ＲＳ２３２Ｃの電文を判定して条件分岐します。
判定する電文は半角１６文字の英数字と記号です。
但し、ダブルコーテーション「”」の記号は入力できません。
受信デリミタは空白文字コード(0x20)以下の 0x00～0x1F のコードで判定します。

２．プログラムボタンの挿入と削除

2-1.ボタンの挿入

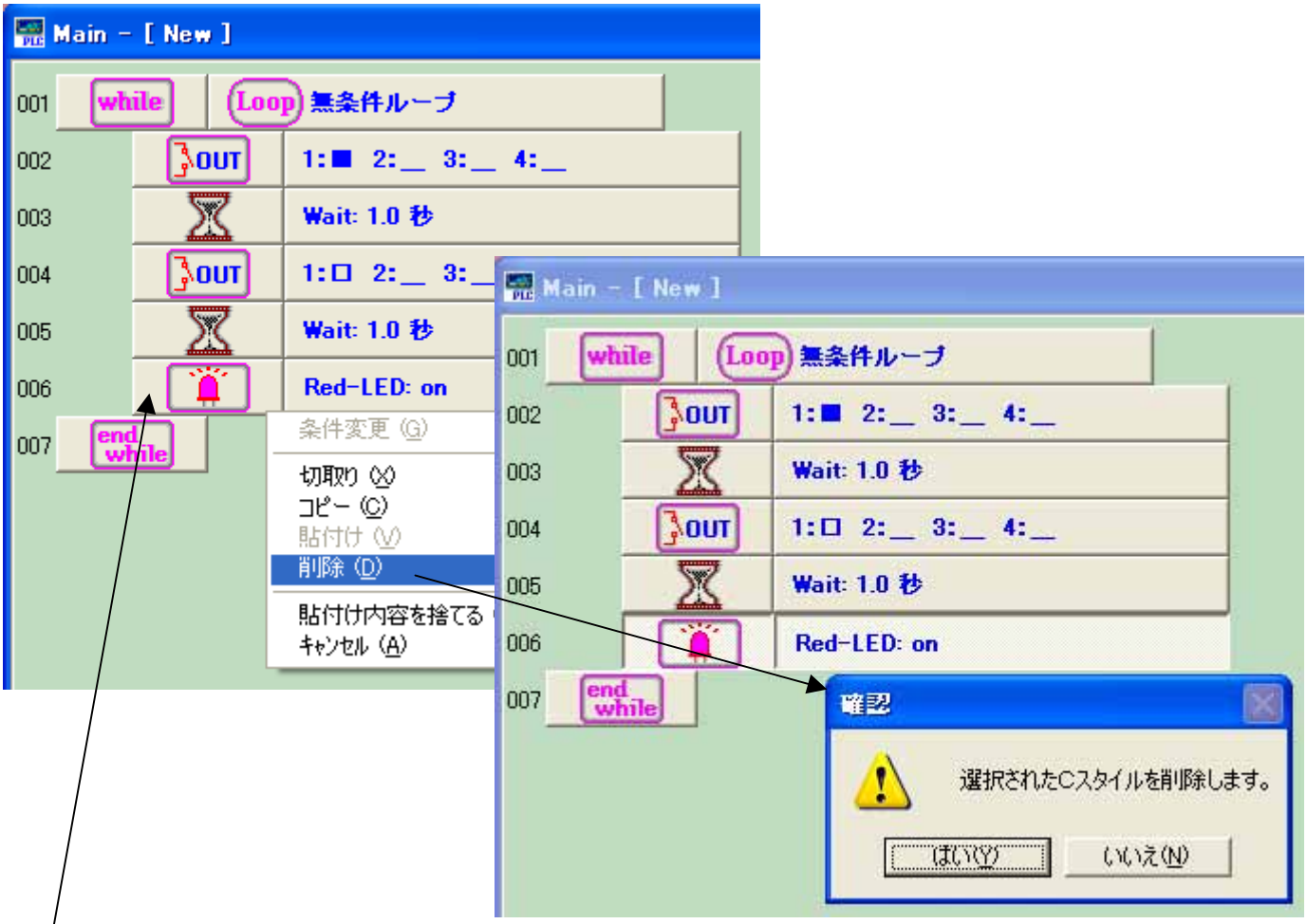
① 先に、挿入したいボタンをプログラムボタンリストから選択しておきます。

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、出力制御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出力制御ボタンが挿入されたところです。

2-2.ボタンの削除



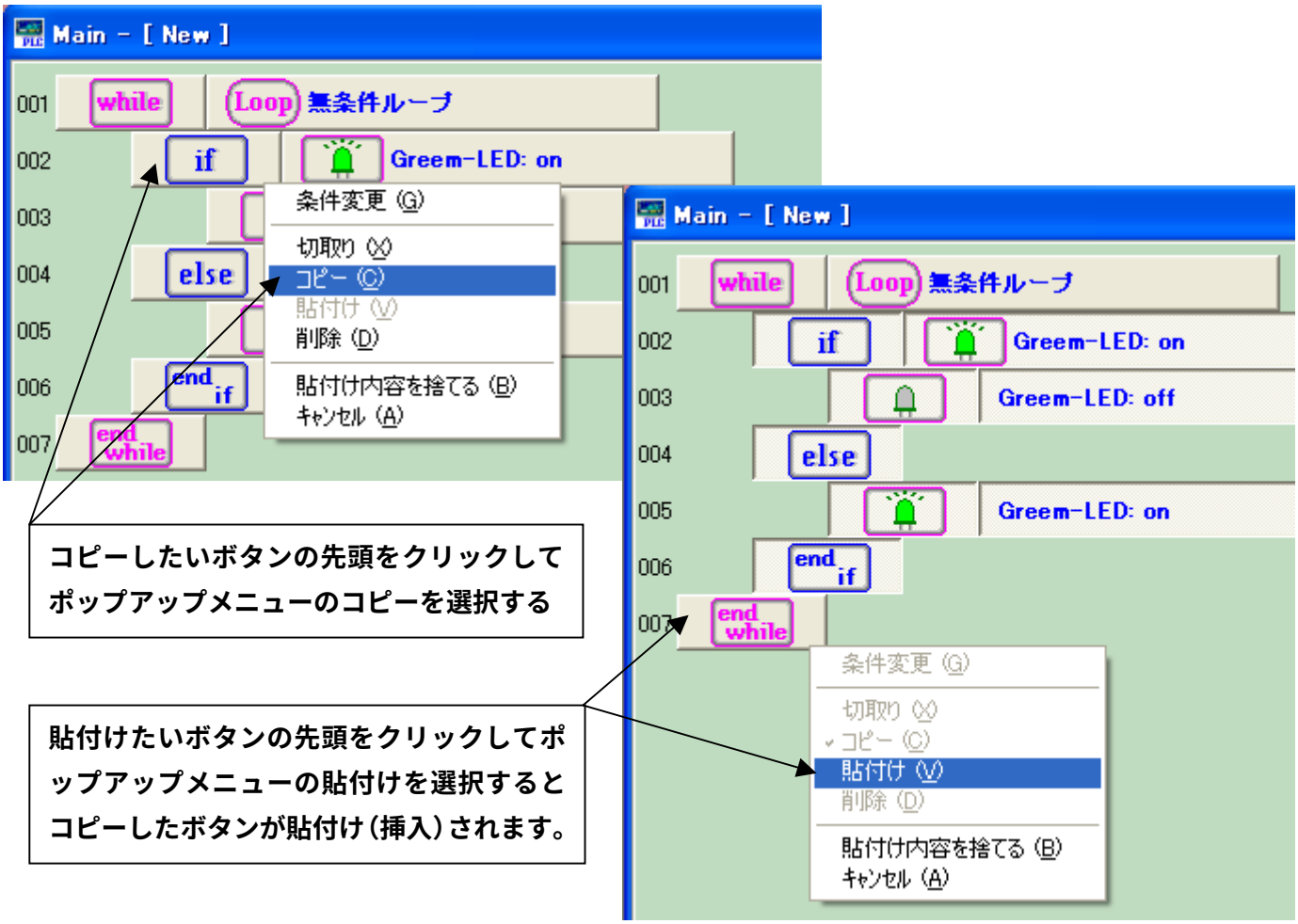
① 削除したい行の前側のボタンをクリックするとポップアップメニューが表示されますので、「削除」を選択します。

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。

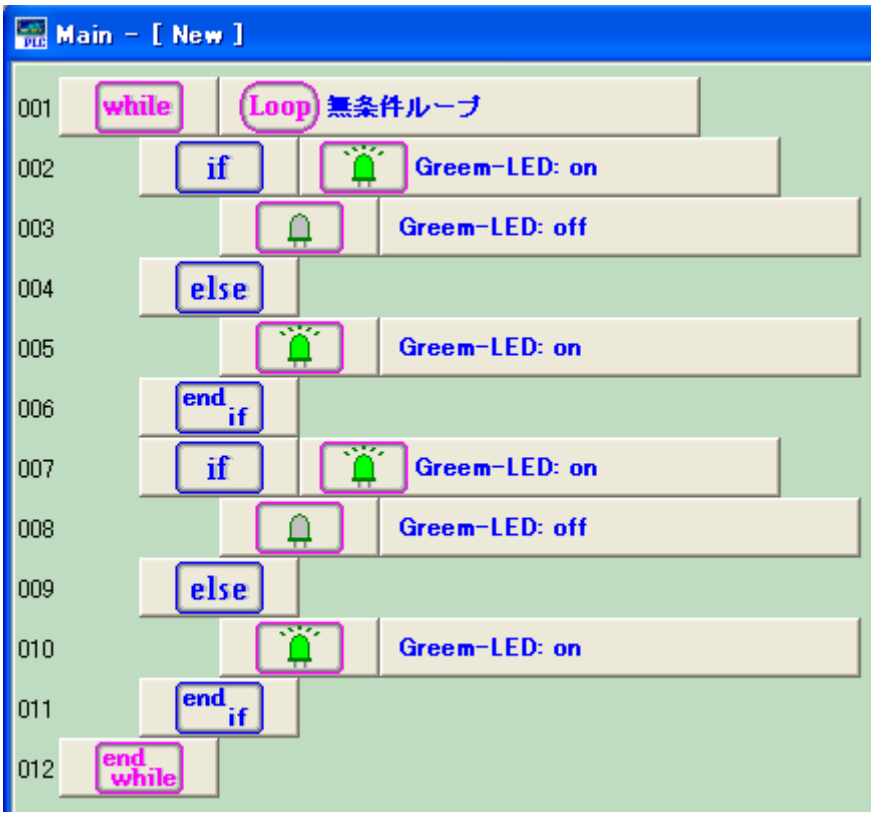
「while」や「for」または「if」ボタンを選択した場合は、「end while」や「end for」または「end if」までの範囲が削除などの対象となります。

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

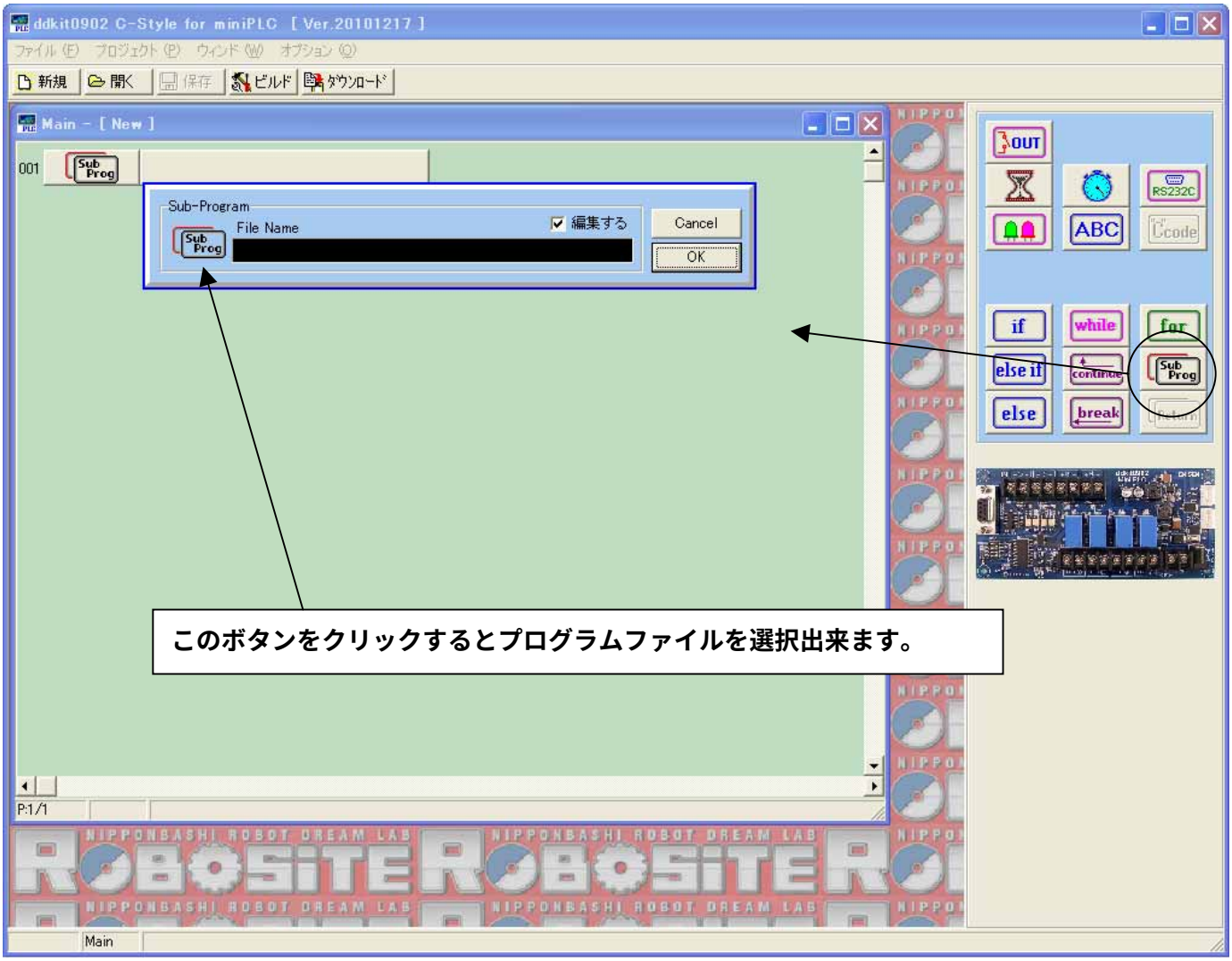
貼付けたいボタンの先頭をクリックしてポップアップメニューの貼付けを選択するとコピーしたボタンが貼付け(挿入)されます。



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いらなくなった場合は、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．サブプログラムの説明

3-1.サブプログラムボタン

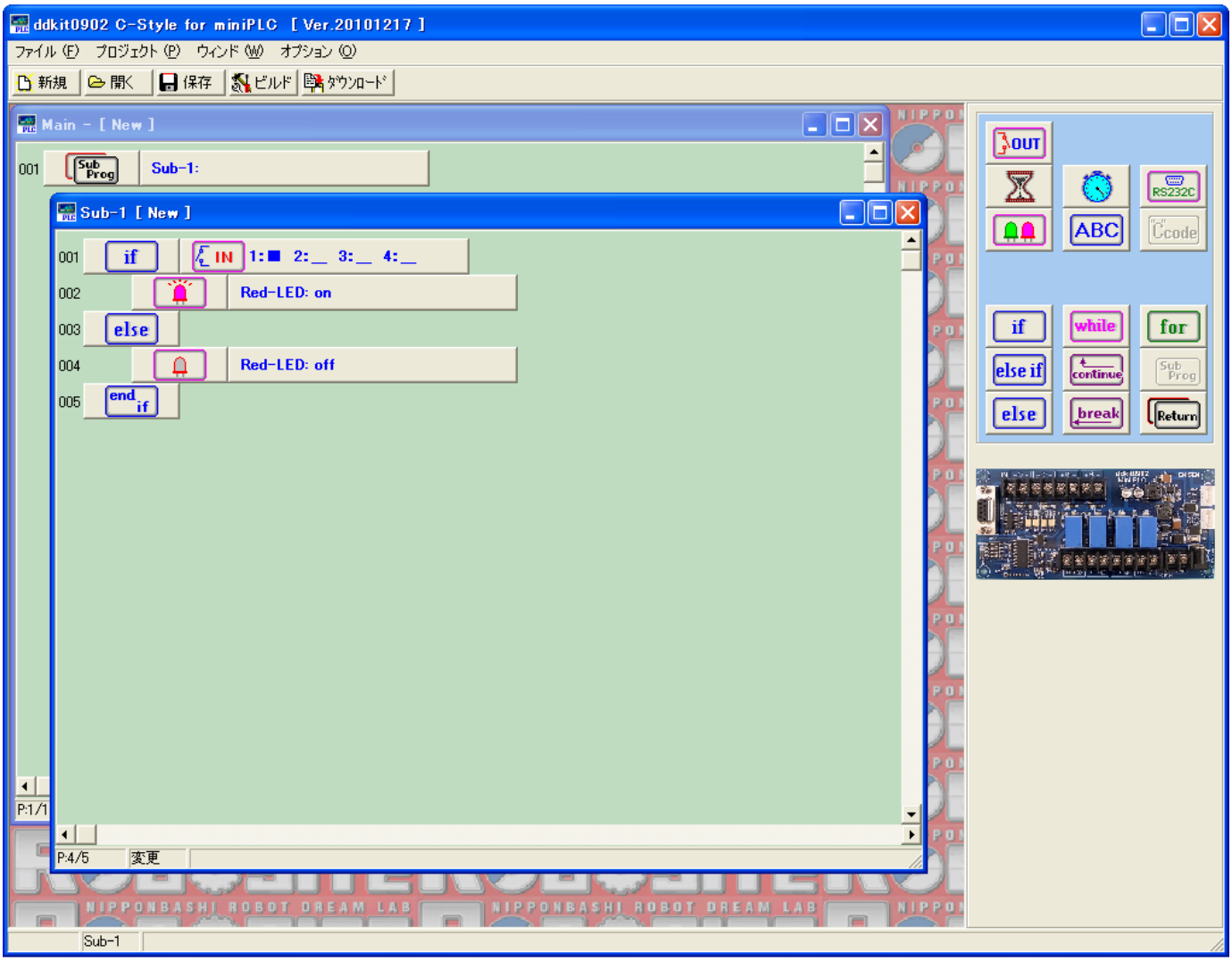


Ｃ－Ｓｔｙｌｅプログラムのサブルーチンを作成します。
同じ処理を何回も再利用したい場合や、プログラムを見やすくする為に、一定の処理プログラムを一つのプログラムボタンにまとめることが出来ます。

サブプログラムは 20 種類作成することが出来ます。
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出来ません。

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、「サブプログラム」ボタンをクリックするとファイルから選択出来ます。

3-2.サブプログラムの編集



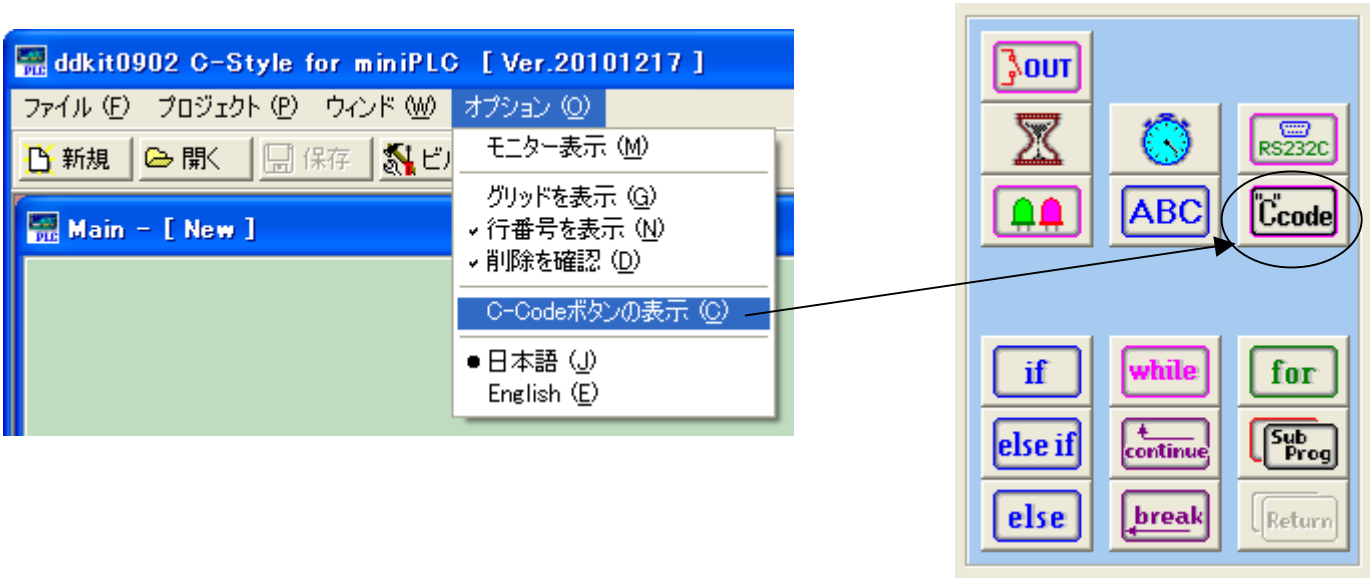
サブプログラムの編集フォームが表示されたら、メインフォーム同様にプログラムボタンリストからプログラムボタンを選択して、サブプログラムを編集して行きます。

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（タイトルバーをマウスでクリックして濃いブルーの表示になった状態）にしてから、「保存」ボタンをクリックします。

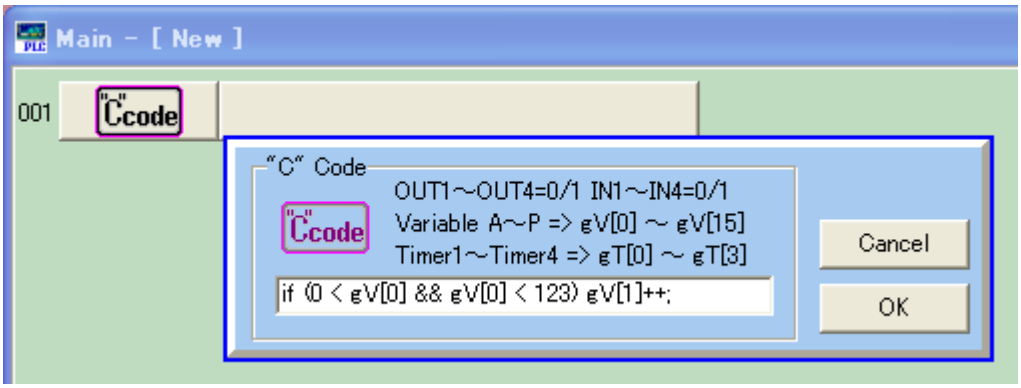
保存したファイル名が、メインフォームのサブプログラムボタンのコメント表示となります。

４．オプションボタン

オプションメニューの「Ｃ－Ｃｏｄｅボタンの表示」を選択すると、プログラムボタンリストにＣ－Ｃｏｄｅボタン機能が有効となります。



C-Codeボタンは通常のC-Styleプログラムボタンでは実現できないＣ言語コードを直接記述することが出来るボタンです。



メモ１

メモ２

⚠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: http://www.daisendenshi.com
e-mail: ddk@daisendenshi.com