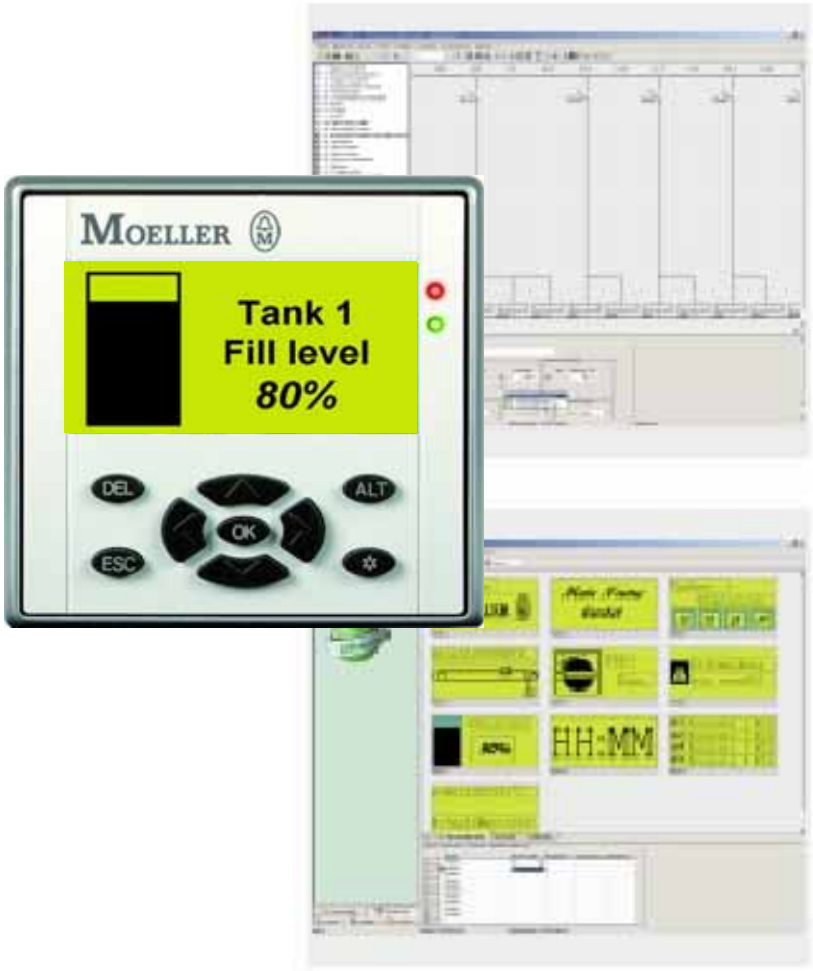
EASY-SOFT　5.10　Pro
（MFD-Titan 用プログラミングソフトウエア）

スタートアップガイド　Ver.0105

ビジュアルプログラム編

本書は MFD−Titan 限定のビジュアル化機能について解説してあります。
すぐに使えるサンプルプログラム解説付き



イートン・エレクトリック・ジャパン株式会社

本社
〒532-0003
大阪市淀川区宮原 3-5-24
新大阪第一生命ビル F8
Tel:06-6150-1281　Fax:06-6150-1285

横浜支店
〒222-0033
神奈川県横浜市南北区新横浜 2 丁目 5 番 9 号
新横浜フジカビル 5 階
Tel:050-3540-6568（IP）045-472-0490
FAX:045-472-0590

三島事業所
〒411-0801
静岡県三島市谷田 61-1
Tel:050-3541-1572（IP）　055-972-1370
FAX:055-972-3840

目　次

1　MFD－Titan でのプログラミングの流れ

MFD－Titan は easy プログラムリレー800 シリーズと同等の機能を有し、さらに高度なグラフィック機能を搭載した新型プログラムリレーです。
MFD－Titan のパソコン上でのプログラムの流れは以下のようになります。

1. Project 定義（使用デバイスの選定、組合せ）
↓
2. Visualization 作成（グラフィック機能）
↓
3. ラダー回路作成
↓
4. プログラムを MFD－Titan 本体にダウンロード
↓
5. プログラム実行

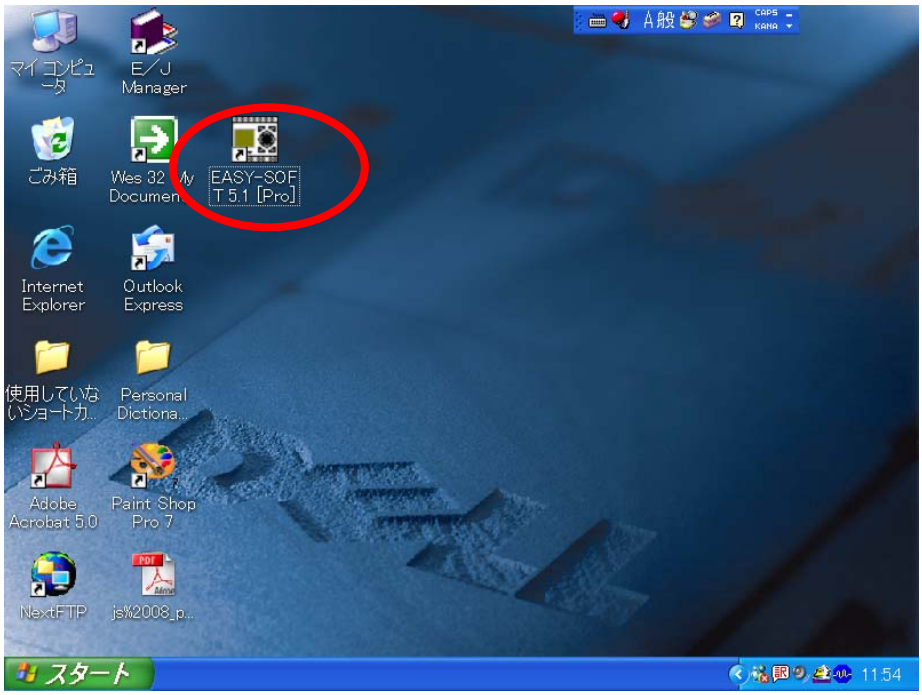
■ 他の easy プログラムリレー（500/700、800 シリーズ）と共通

☀ MFD－Titan に特有の機能

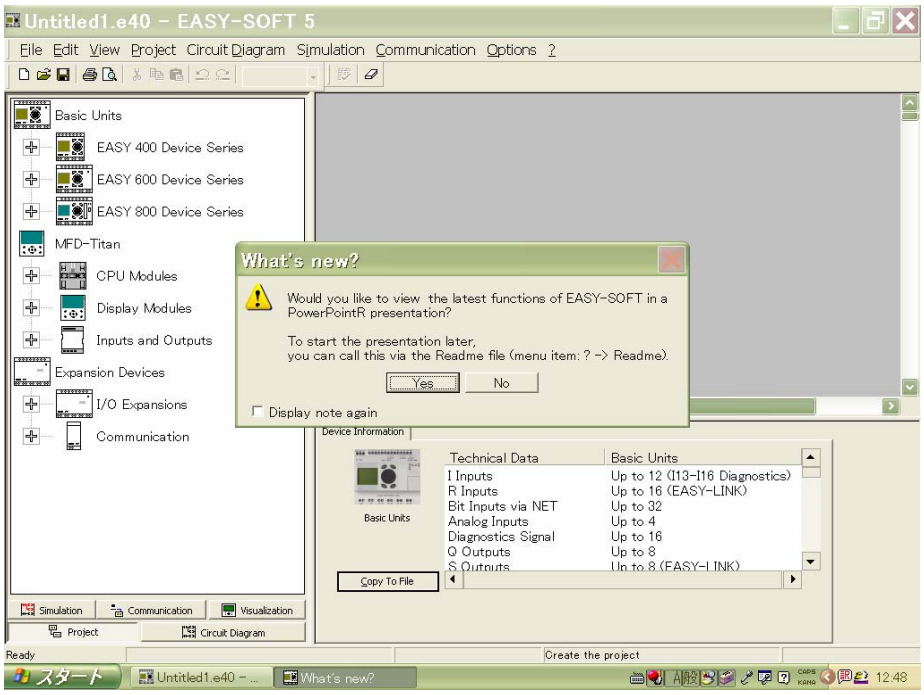
次ページ以降では、点滅警報マークやスイッチをアニメ化、キーパッドでのデータ入力、メニュー画面の作成など、システム環境を最適にするソリューションを簡単に学んでいただけます。これにより MFD－Titan のエッセンスをご理解いただけ、お客様独自のプログラムを作っていただけるようになっています。

２．EASY　SOFT　5.10　Pro の立ち上げ

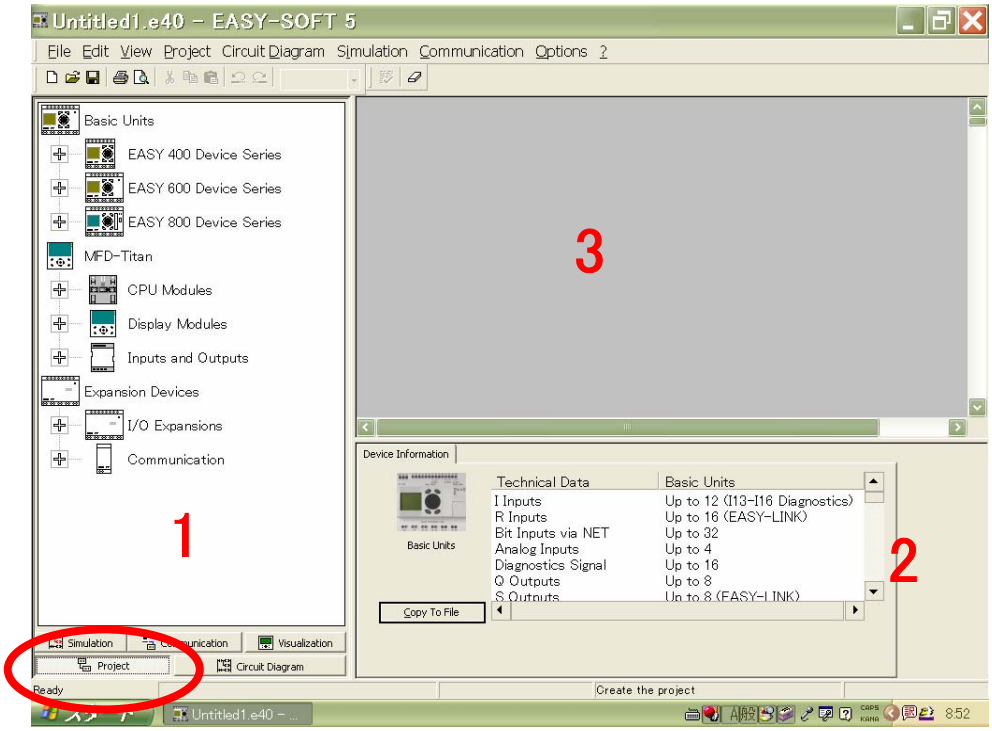
easy ソフトをインストールすると、パソコンのデスクトップ上に EASY　SOFT　5.10　Pro のショートカットアイコンができます。



このアイコンをダブルクリックします。



新機能説明のパワーポイントショーを実行しますか？と聞いてきますので、「No」をクリックします。

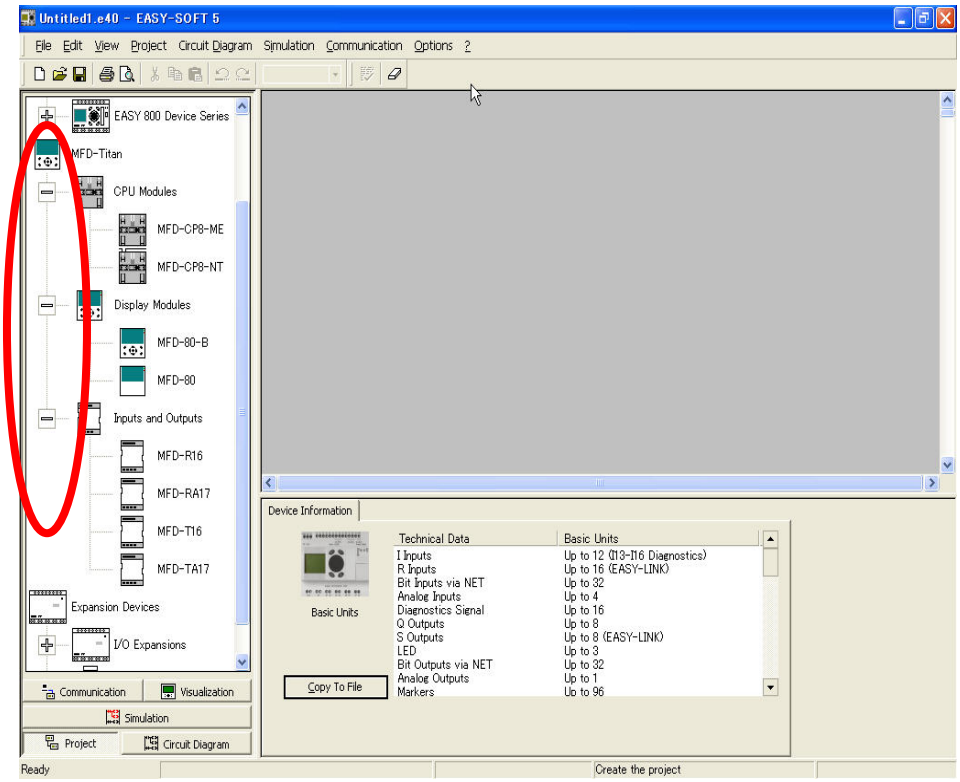


これが EASY　SOFT　5.10　Pro の最初の画面です。
画面は3つの部分から構成されていて、それぞれ以下のような呼び名がついています：
1：ツールボックス
2：プロパティフィールド
3：ワークベンチ
です。
通常は様々なツールを1：ツールボックスからドラッグしてきて、3：ワークベンチで作業します。プログラミングに関する設定等を、2：プロパティフィールドで行います。

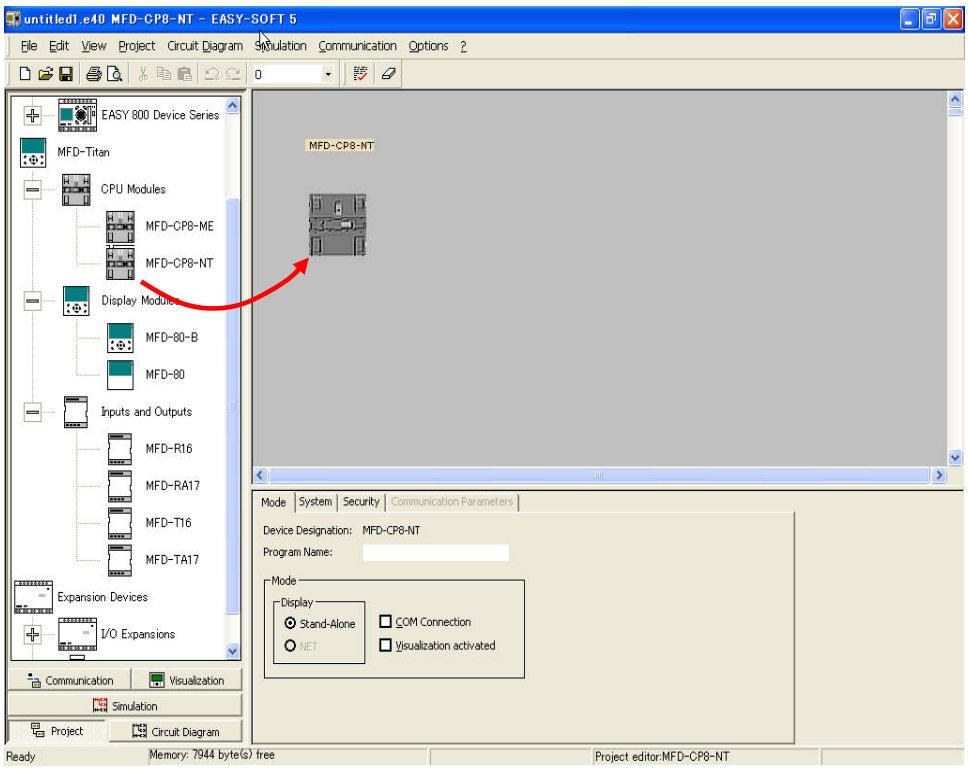
ツールボックス下段の「Project」がクリックされている状態を確認してください。

この状態で次節の Project 定義（使用デバイスの選定）を行います。

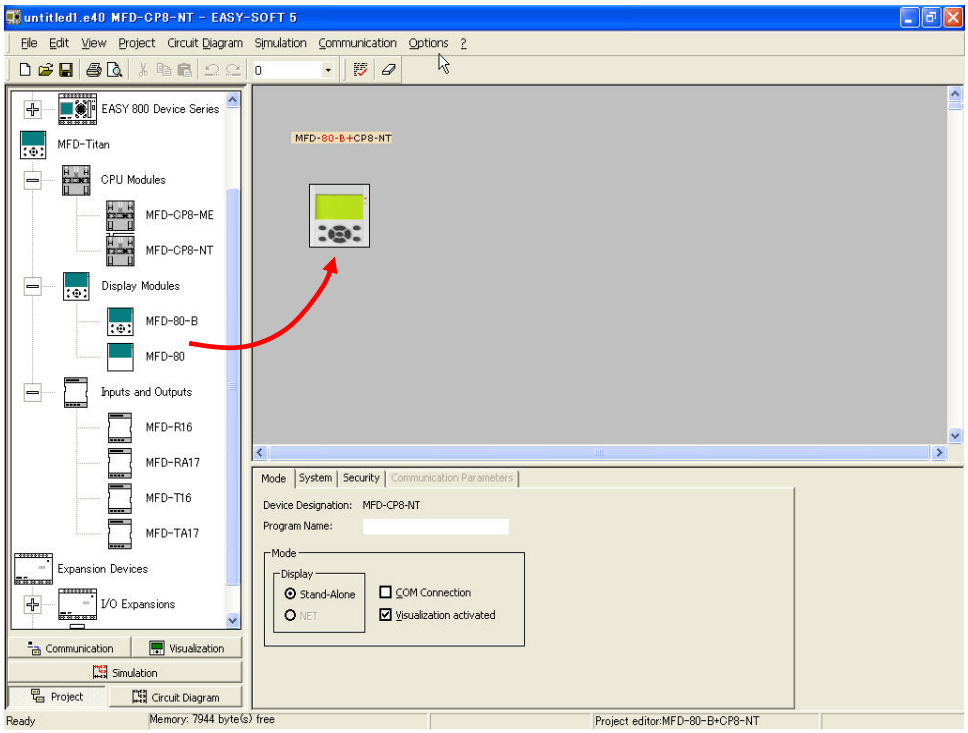
３． Project 定義（使用デバイスの選定）



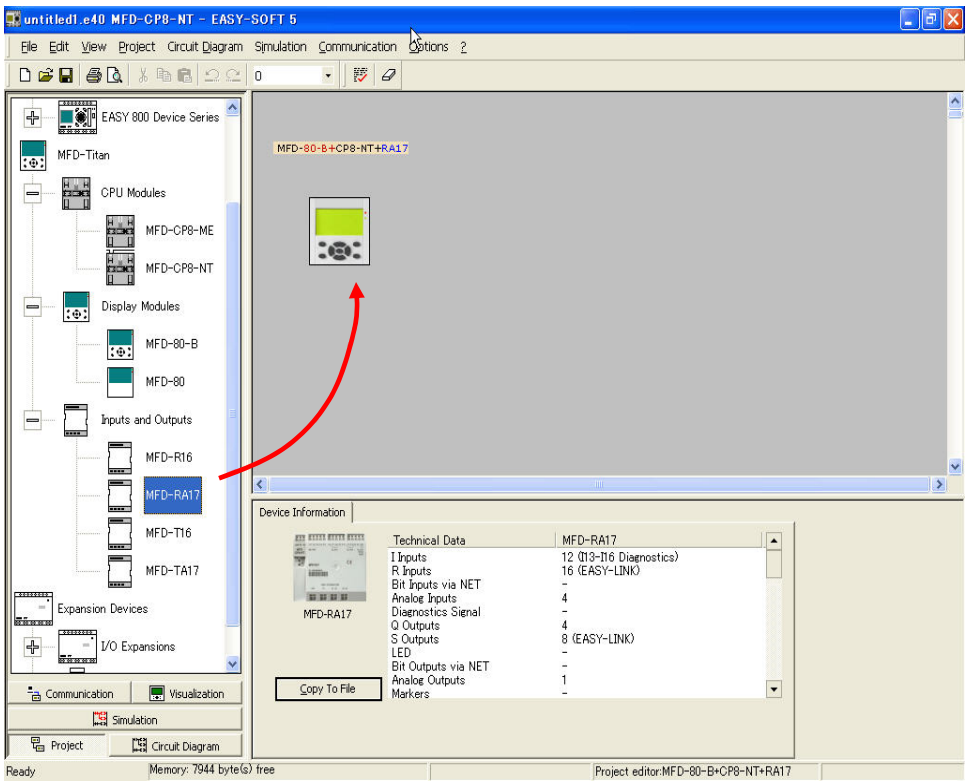
MFD－Titan の下の＋マークをクリックし、各形式をドロップダウンさせます。



まず、CPU を選択します。ここでは MFD-CP8-NT をクリックして、右側のワークベンチへドラッグします。



同様に、Display Module も選択しましょう。ここでは MFD-80-B を選び、先ほどワークベンチへドラッグした CP8-NT 上へ重ねます。

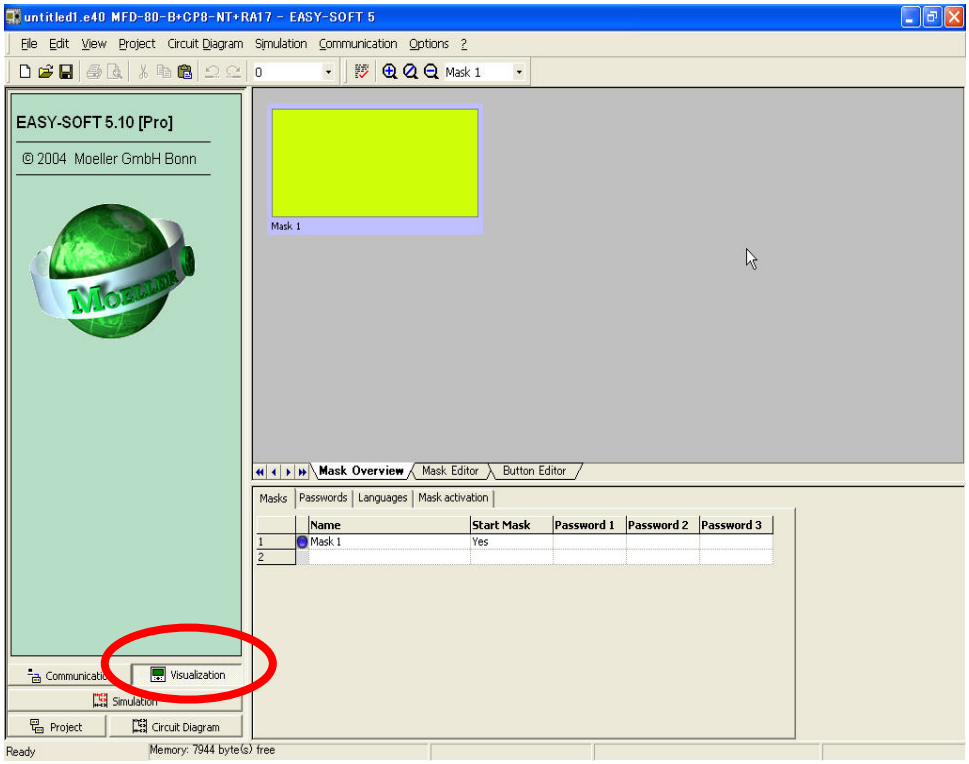


最後に I/O ユニットを選択します。ここでは MFD-RA17 を選択し、先ほどまでの CP8-NT&MFD-80-B に重ねます。これで Project 定義（使用デバイスの選択）が完了し、次節からの実際のプログラミングの準備ができました。

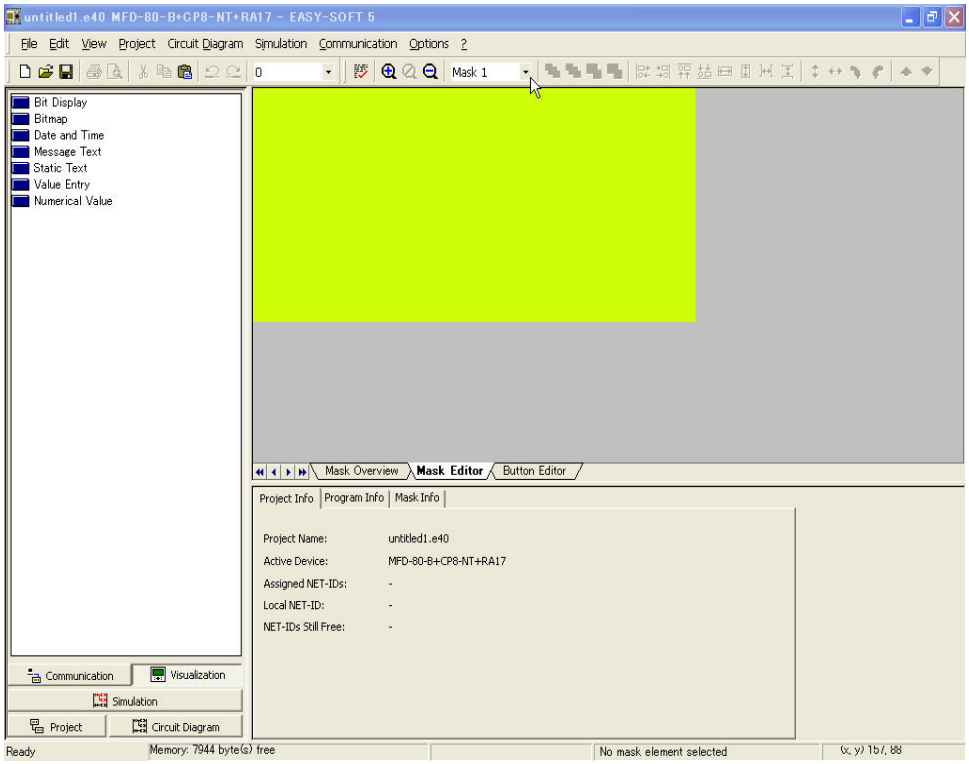
４． 点滅警報ビットマップの作成

ここでは以下のようなプログラムを作ってみます。

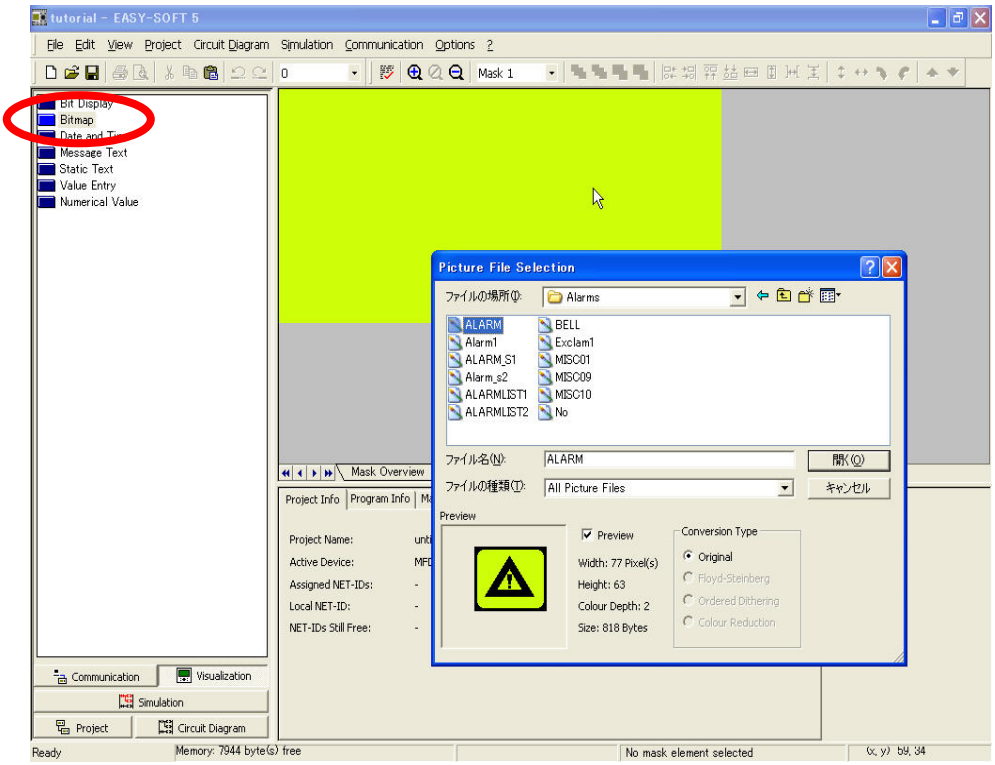
[タスク定義] I1 が投入されると、危険を知らせる警報ビットマップが点滅します。I1 がオフの時には画面には何も映りません。



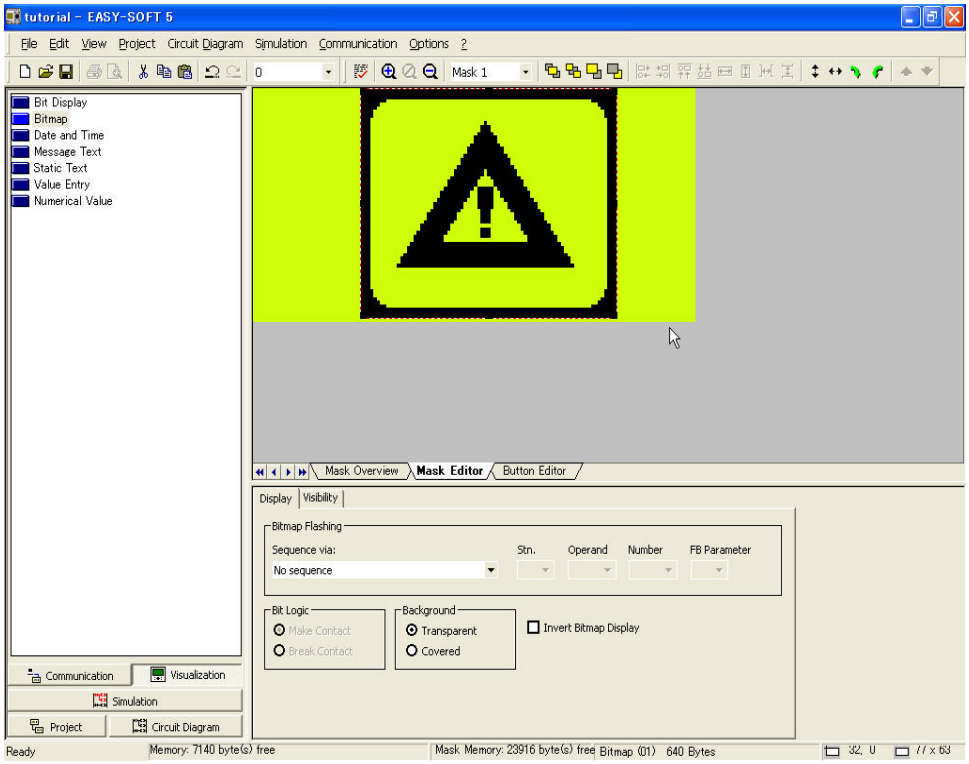
ツールボックスの左下、「Visualization」をクリックすると、上画面が現われます。
「Mask１」と名前の付いた MFD の画面がワークベンチにありますが、これは MFD の画面の１コマに相当し、「マスク」と呼びます。



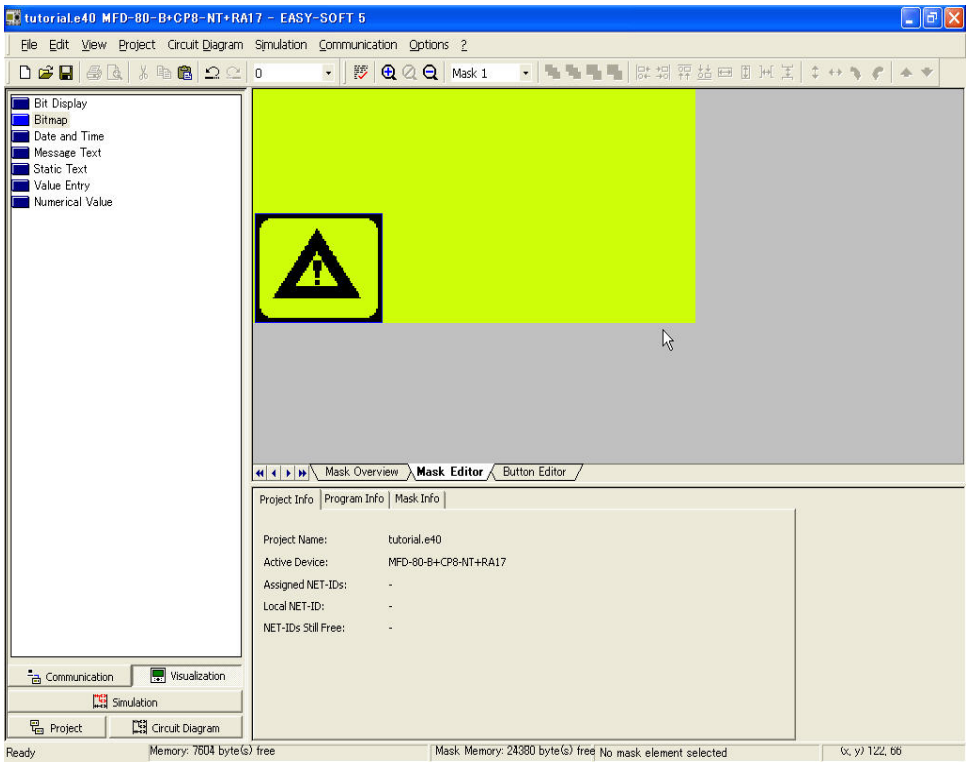
マスクをダブルクリックして、画面を大きくしてください。



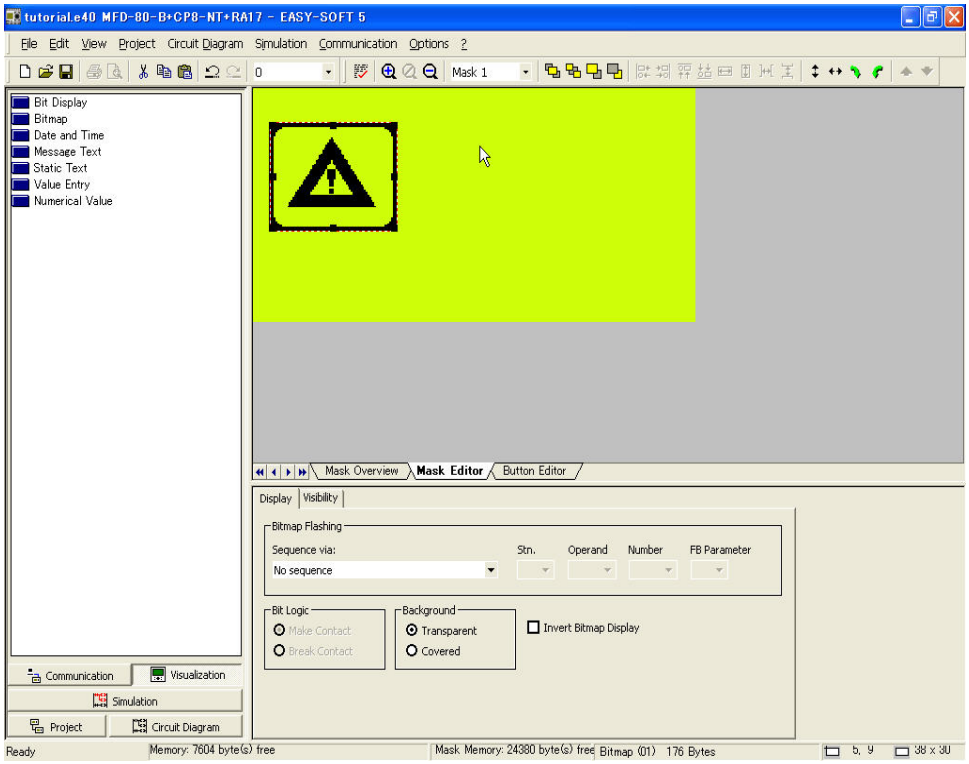
ツールボックスにある Bitmap をクリックして、マスク上にドラッグすると、挿入したいビットマップ画像を選択するウィンドウ（Picture File Solection）が開きます。「Alarms」フォルダ内の「ALARM」を選択してください（Preview で確認できます）。



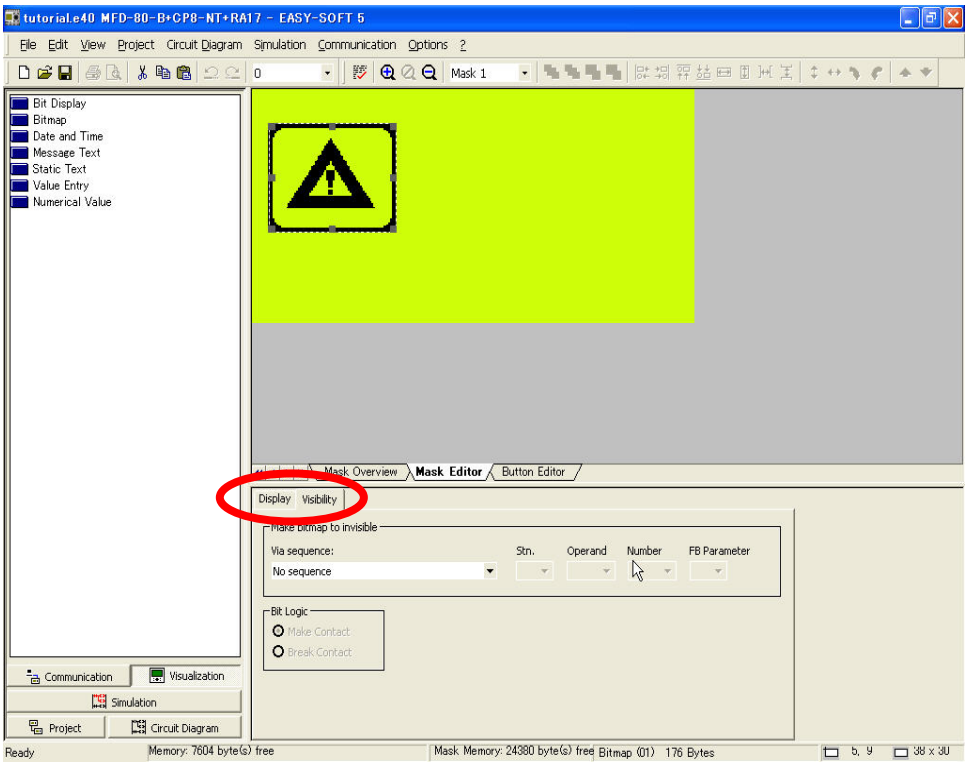
「開く」をクリックすると、ビットマップがマスク上に現われます。



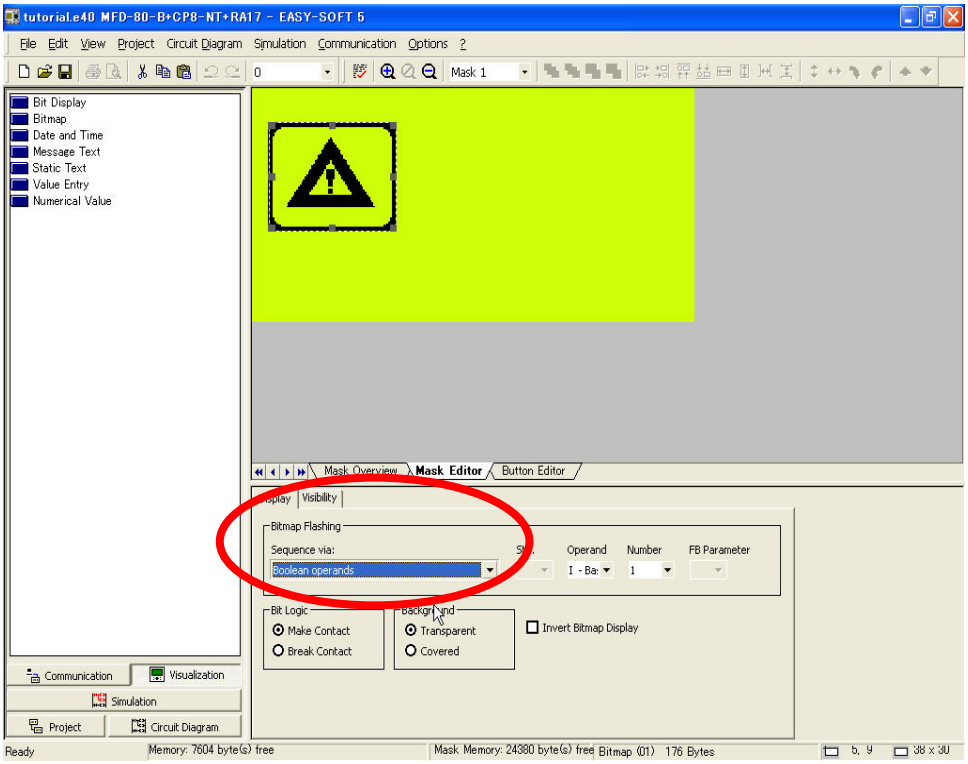
ビットマップの 4 隅か辺の中心をポイントすると、ポインタが両方向矢印に変わります。これをドラッグして大きさの調節をしてください。



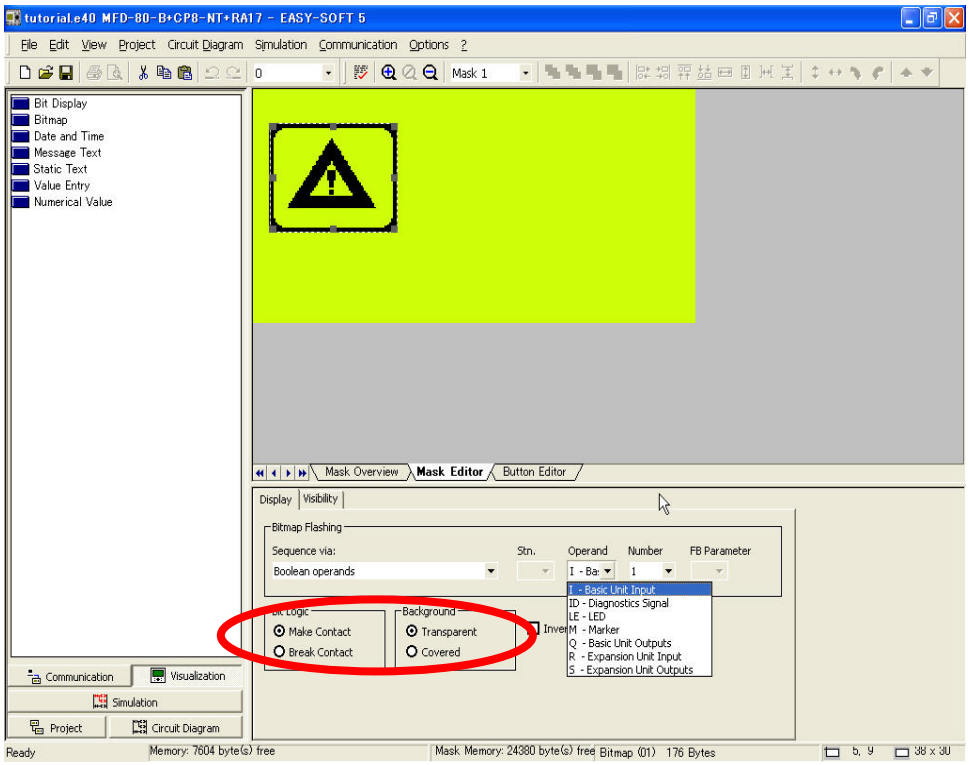
さらに、ビットマップ内部をポイントすると、ポインタが4方向の矢印に変わります。この時にドラッグをすると、ビットマップの位置を変えることができます。



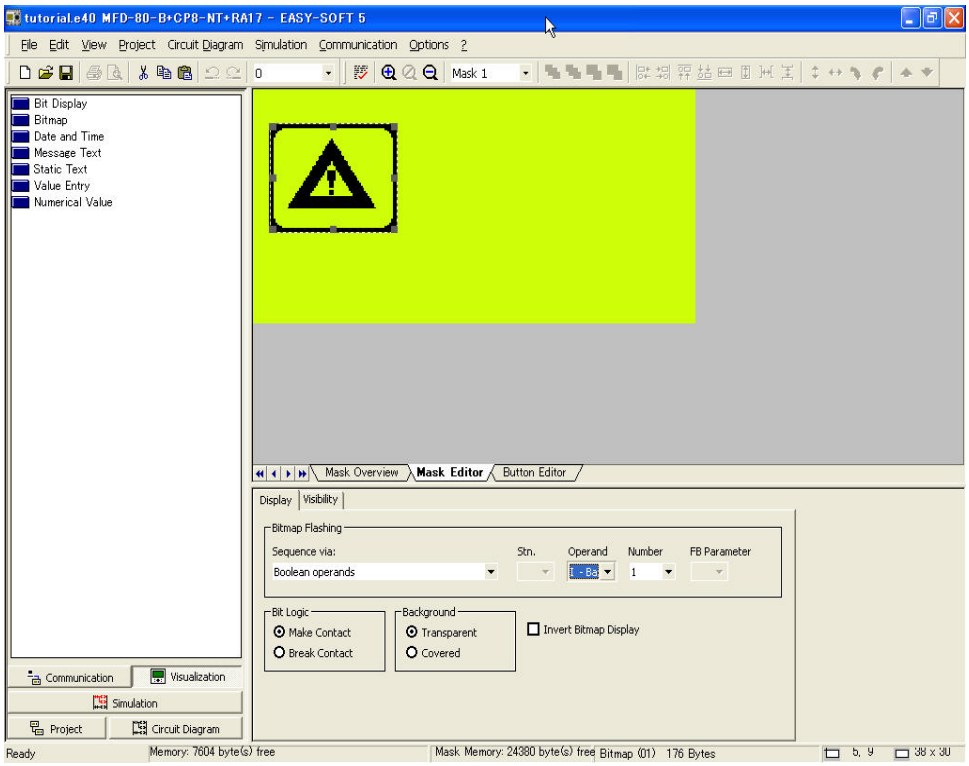
次にビットマップの動作を設定します。ビットマップを選択した状態（赤白の縞線がビットマップを囲んだ状態）で、2：プロパティフィールドの Display をクリックします。



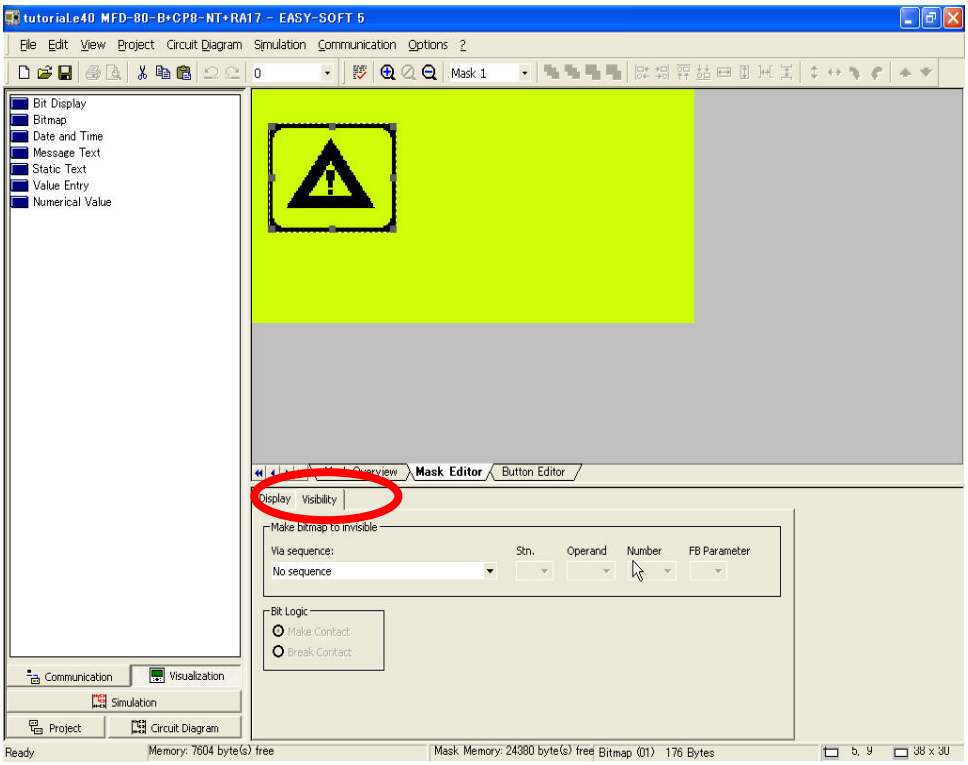
Bitmap　Flashing（ビットマップの点滅）の枠内でどのように点滅させるか設定しましょう。
Sequence　via のメニューから「Boolean　operands」を選びます。
参考：Boolean　operands は「論理式」の意味です。



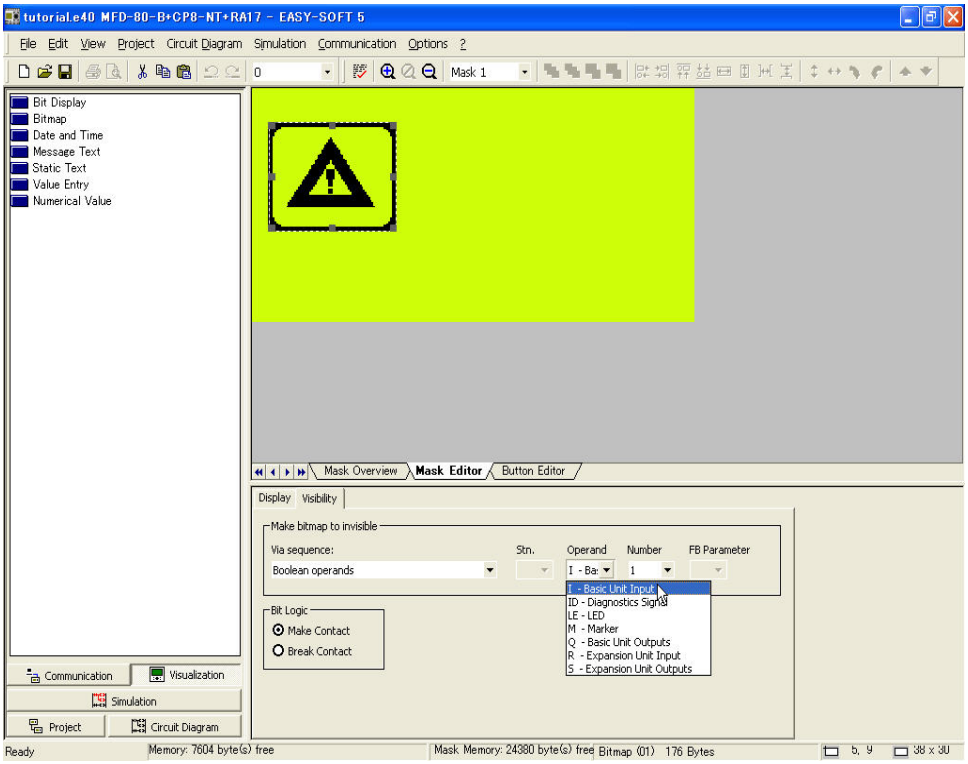
Operand 欄では「I-Basic Unit Input」を選択。
Number 欄では「1」を選択。
Bit Logic の枠では「Make Contact」を選択。



Background の枠では「Transparent（透明）」を選択。
以上の設定は「基本ユニットの入力 I1 が閉じる（メイク接点）と、ビットマップがフラッシング（点滅）します」という意味を持っています。

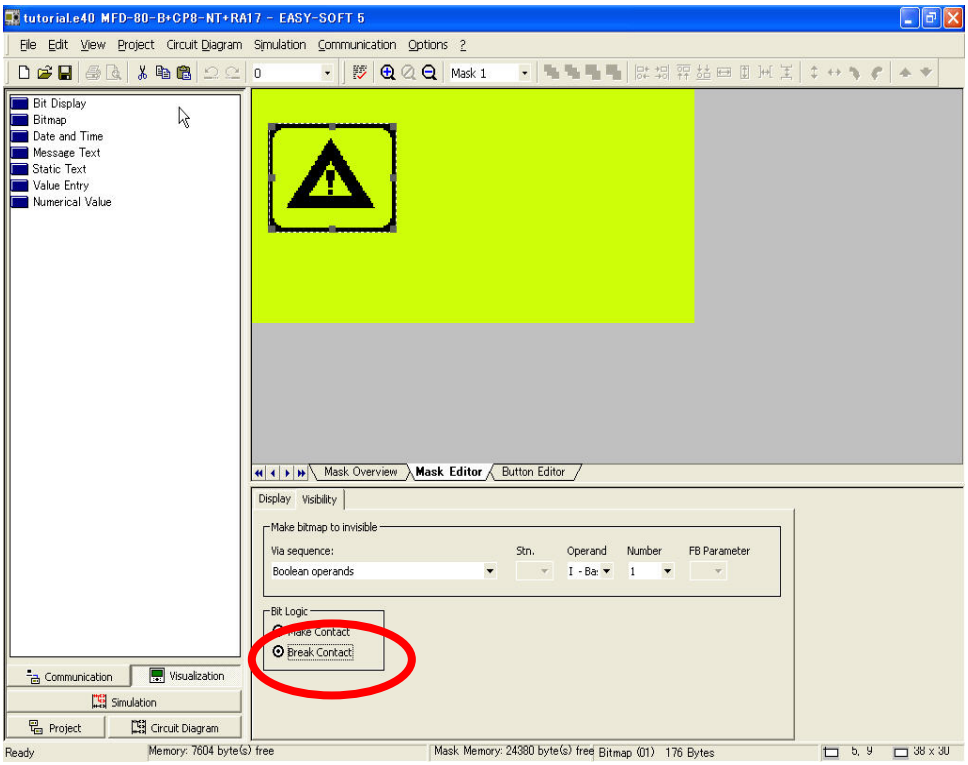


次に I1 がオフの時にビットマップが消える設定を行います。
Visibility タブをクリックしてください。

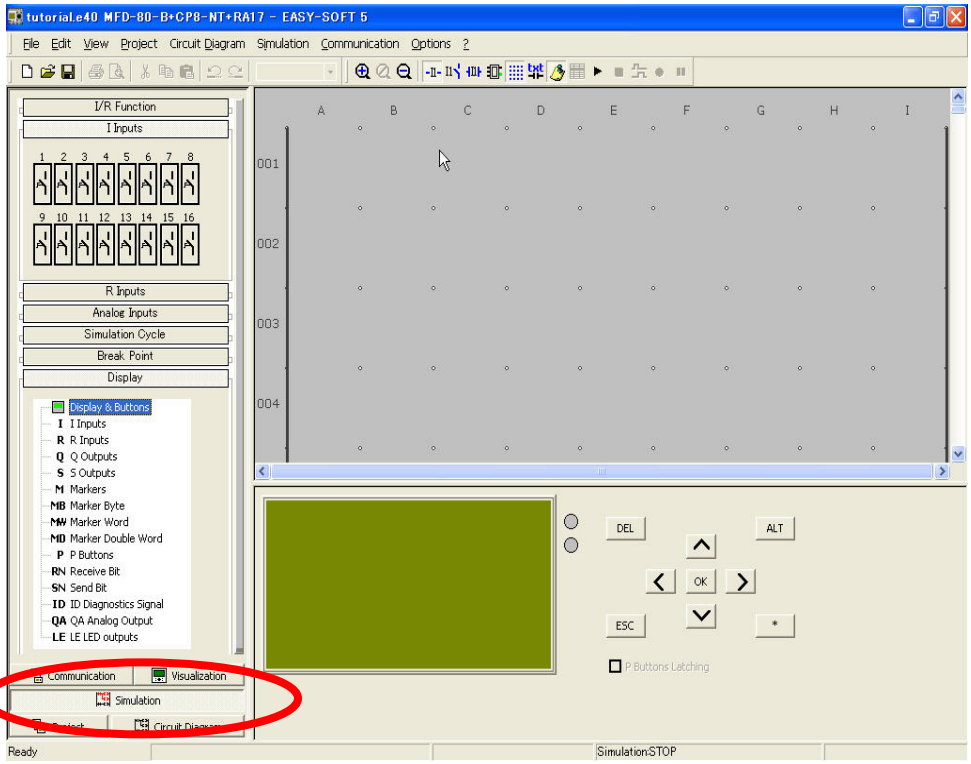


「Make bitmap to invisible（ビットマップを不可視にする）」の枠内の Via sequence で「Boolean operands」を選択、Display タブと同様に、Operand 欄では「I-Basic Unit Input」を選択、Number 欄では「1」を選択してください。

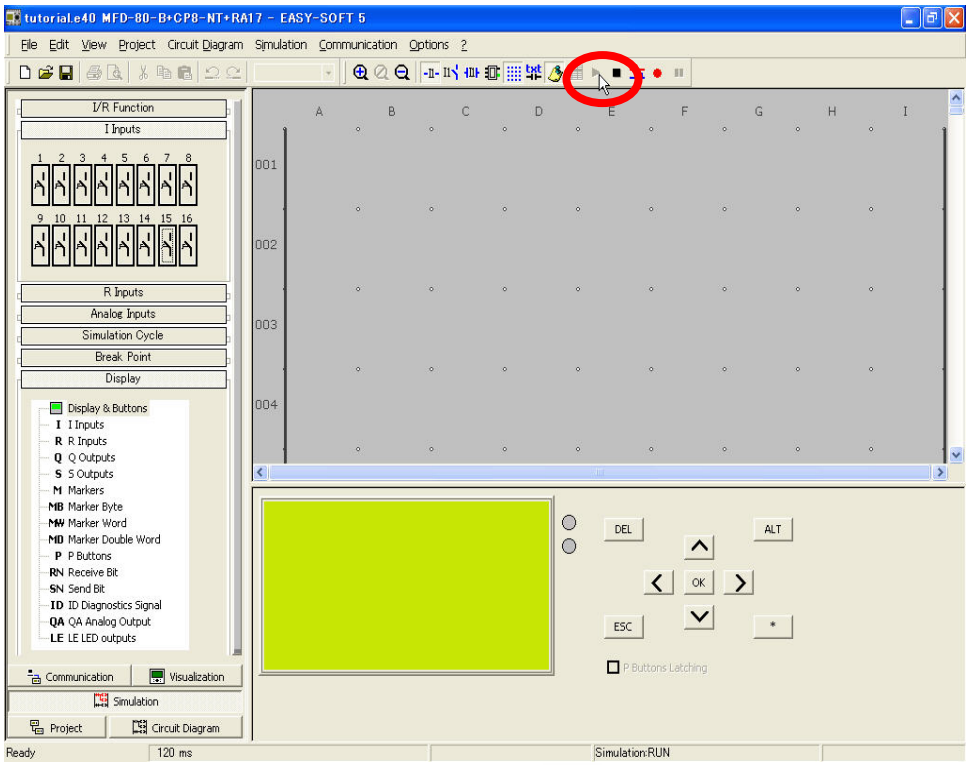
参考：ここで「No Sequence」を選ぶと、I1 がオフでもビットマップが静止状態で表示されます。



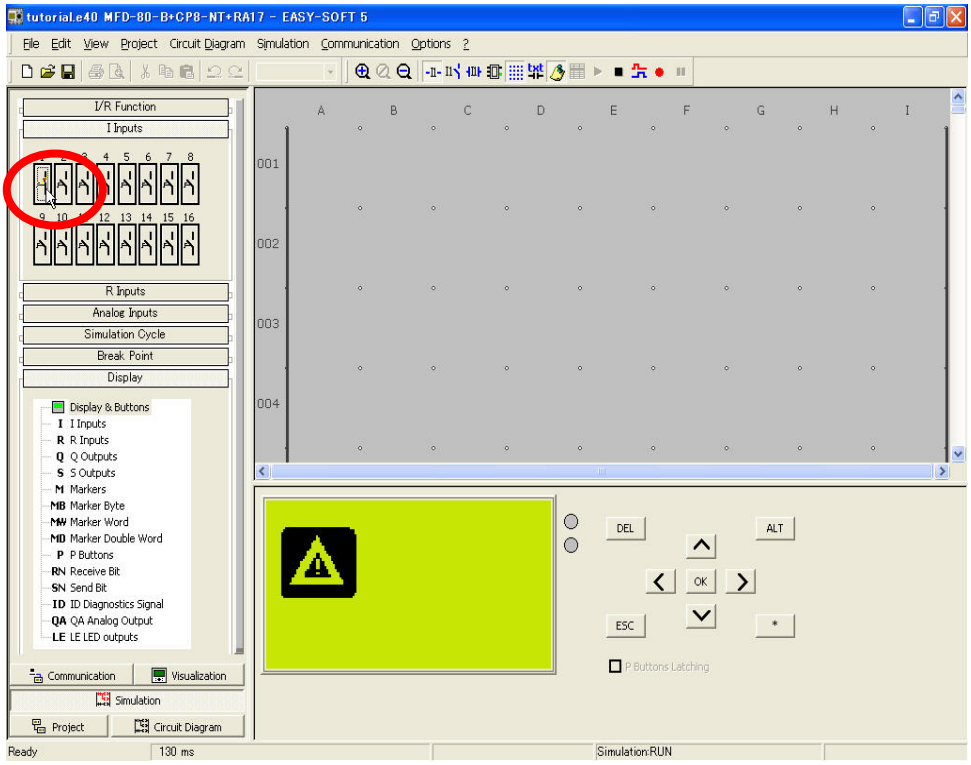
Bitlogic では、今度は「Break　Contact」を選択します。以上は、入力 I1 がオフの時は警報ビットマップを消した状態にする設定です。
これでビットマップ作成は完成です。



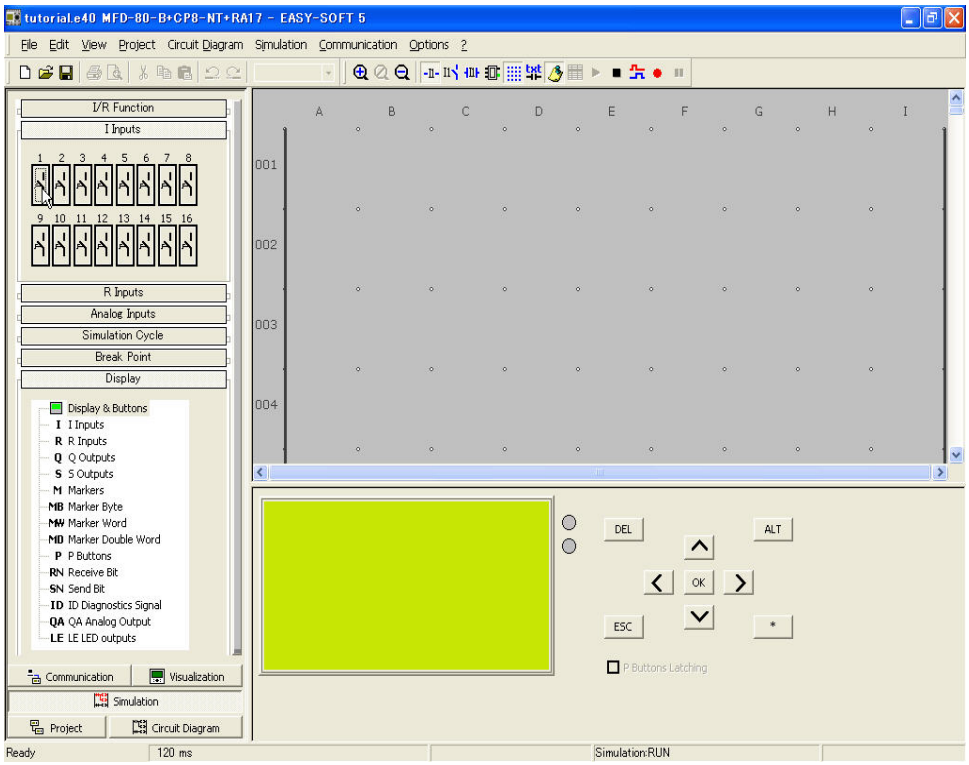
ビットマップが動作するかどうかをシミュレーションしてみましょう。ツールボックス下の「Simulation」ボタンをクリックしてください。
「I Input」ボタンをクリックして、入力ボタンを表示します。
「Display」ボタンをクリックして、一番上の「Display&Button」を選択します。2：プロパティフィールドに MFD のディスプレイとボタンがシミュレーション画面として表示されます。



シミュレーション実行ボタン（Start　Simulation）▶ を押します。画面が明るくなり、シミュレーションが始まります。

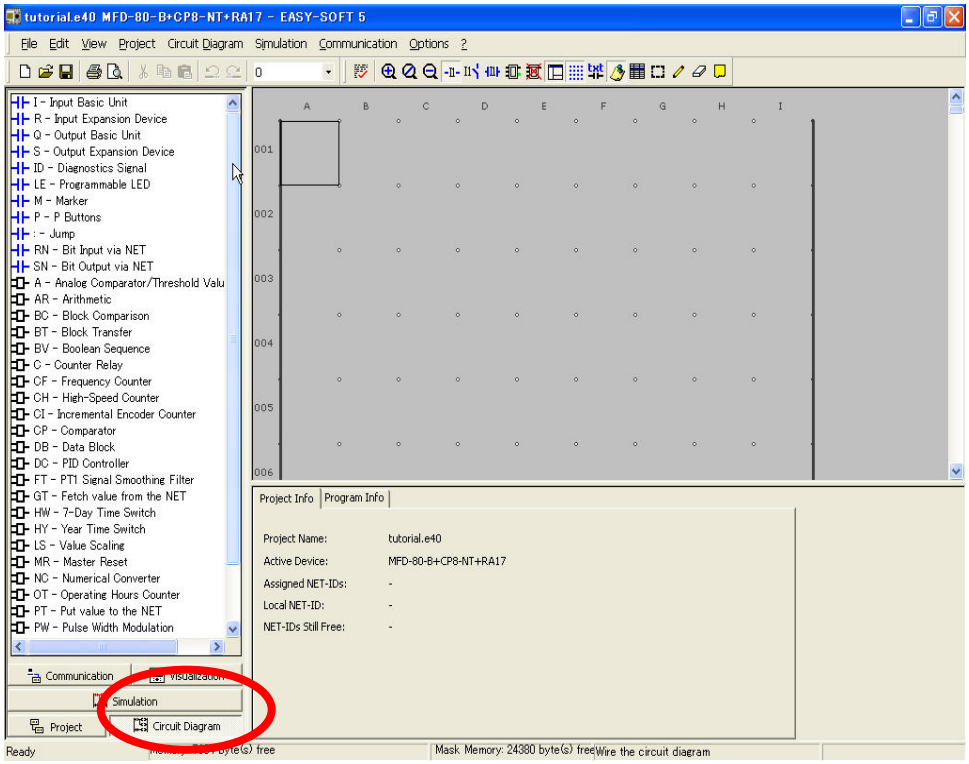


I1 接点が閉じられる（ボタンを押す）と、設定した警報ビットマップが反転点滅で表示されます。

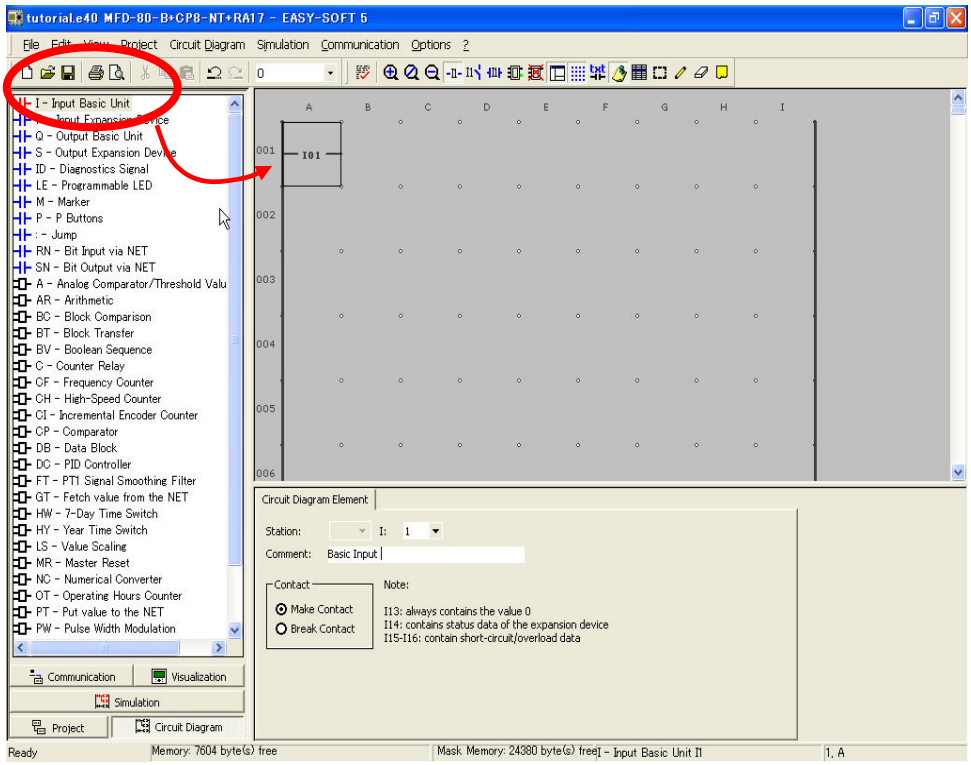


I1 が開かれる(オフになる、再び I1 ボタンを押す)と、ビットマップは消えた状態になります。
これでビットマップが正常に動作することがわかりました。

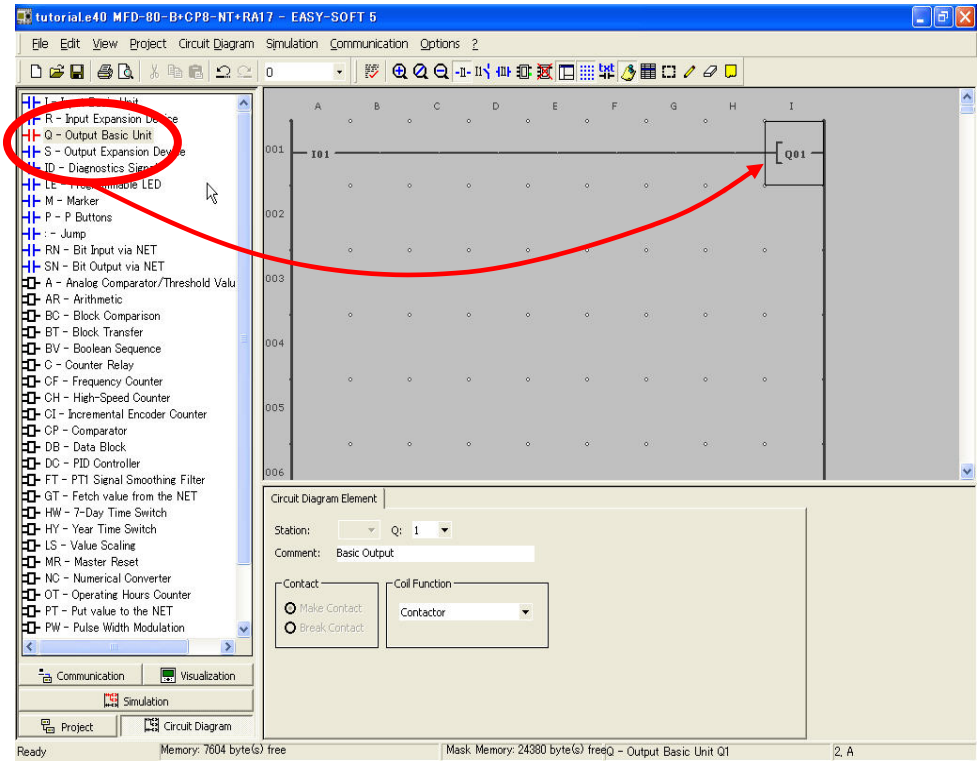
次に、MFD 本体を動かすための、このビットマッププログラム用のラダー回路を作成しましょう。



ツールボックス下の「Circuit　Diagram」をクリックしてください。ワークベンチにラダー回路入力画面が表示されます。

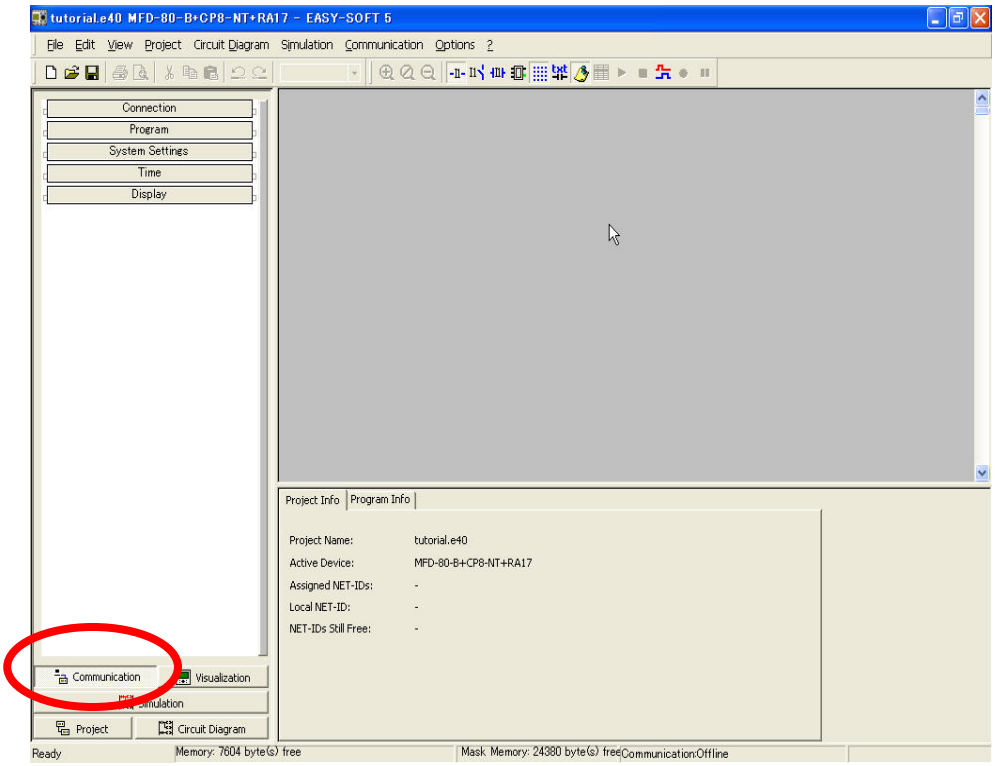


ツールボックスの I-Input Basic Unit をクリックして、ワークベンチにドラッグ。自動的に I1 が挿入されます。I の番号はプロパティフィールドで選択来ますが、ここではこのままにしましょう。

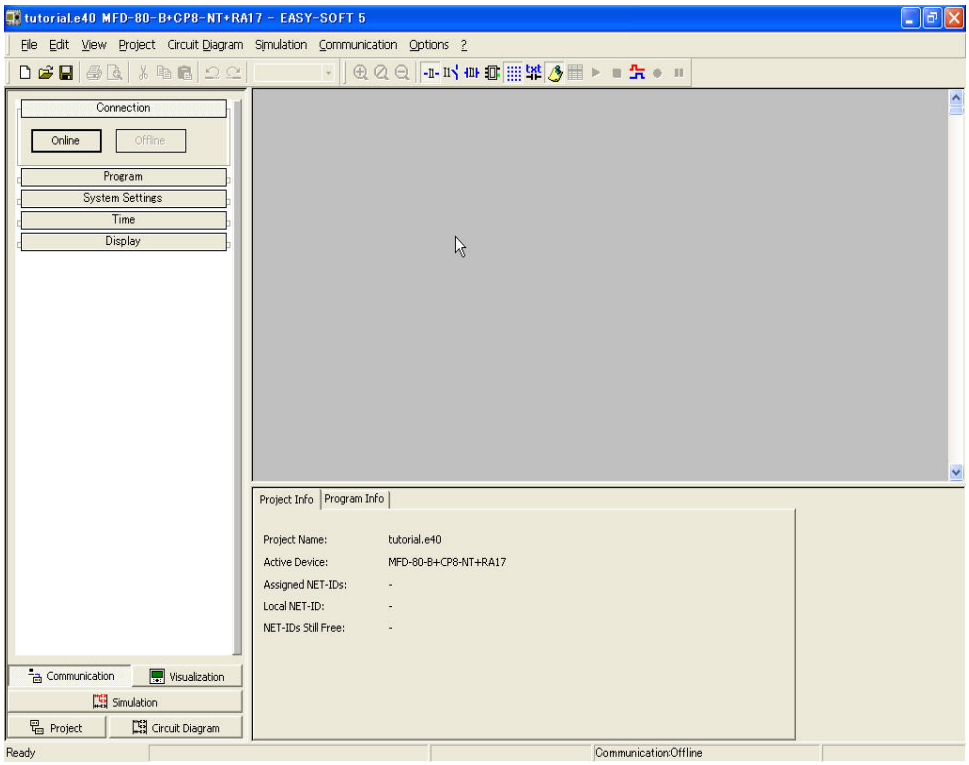


同様に、Q-Output Basic Unit をラダー回路図内の列 I(一番右側)にドラッグすると、I1 からの接続配線が自動で描かれます。プロパティフィールドの設定は Q の番号が1Coil Function が Contactor になっているので、このままにします。
回路図が出来上がりました。以上でプログラムの完成です。

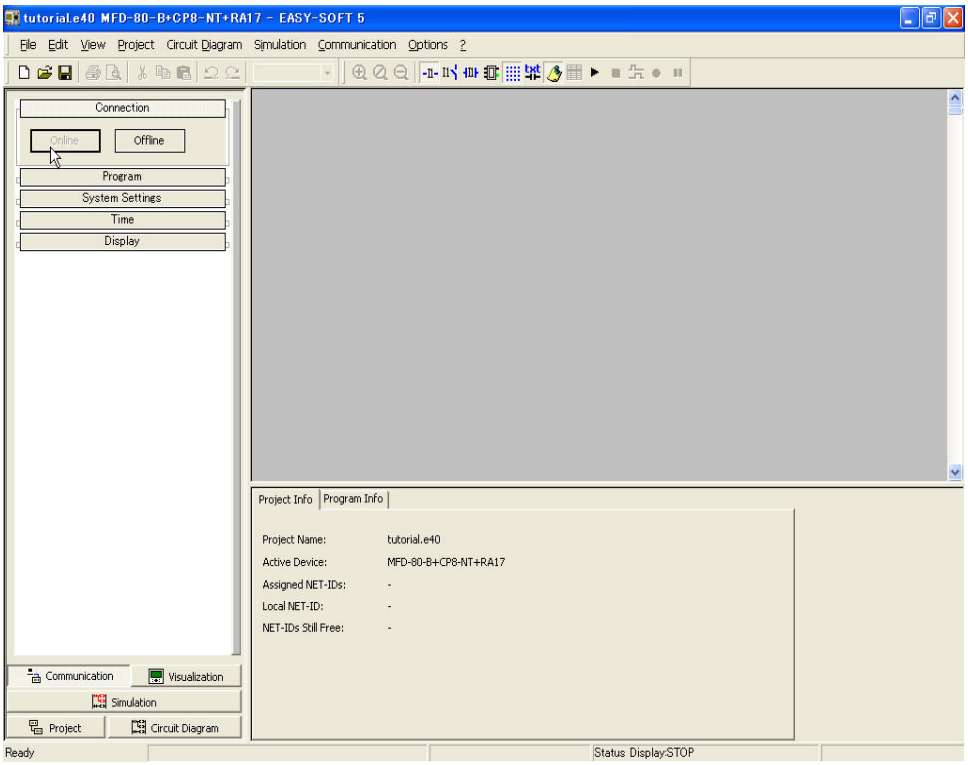
最後に、完成したプログラムを MFD 本体にダウンロードします。
パソコンと MFD 本体が専用ケーブルで正しく接続されていることを確認してください。



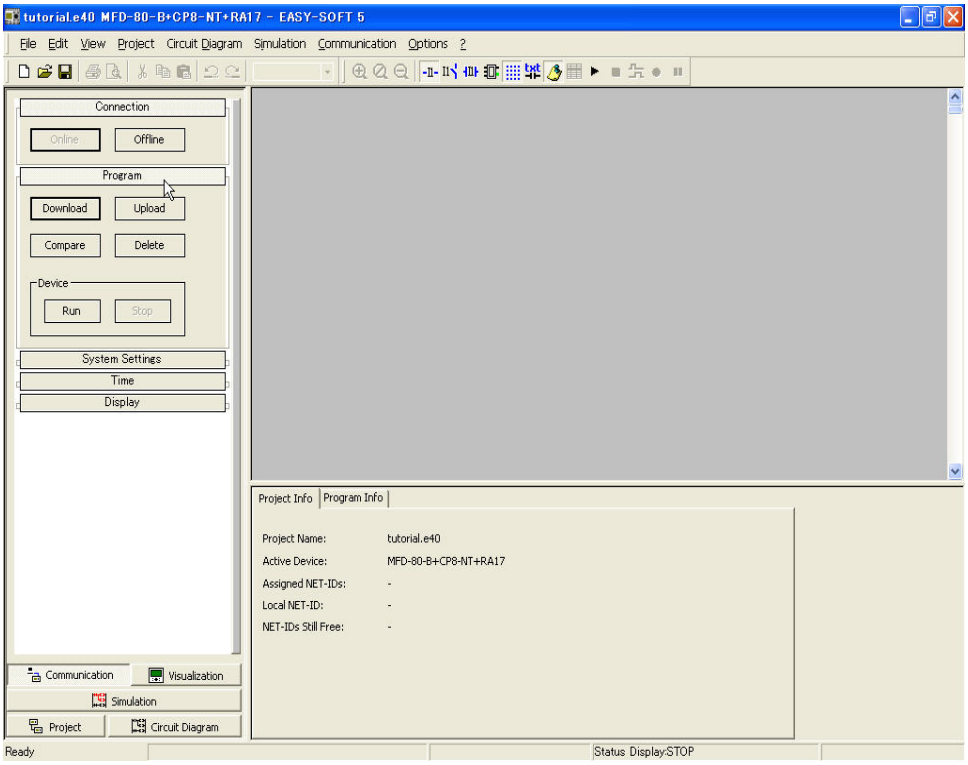
「Communication」ボタンを押します。



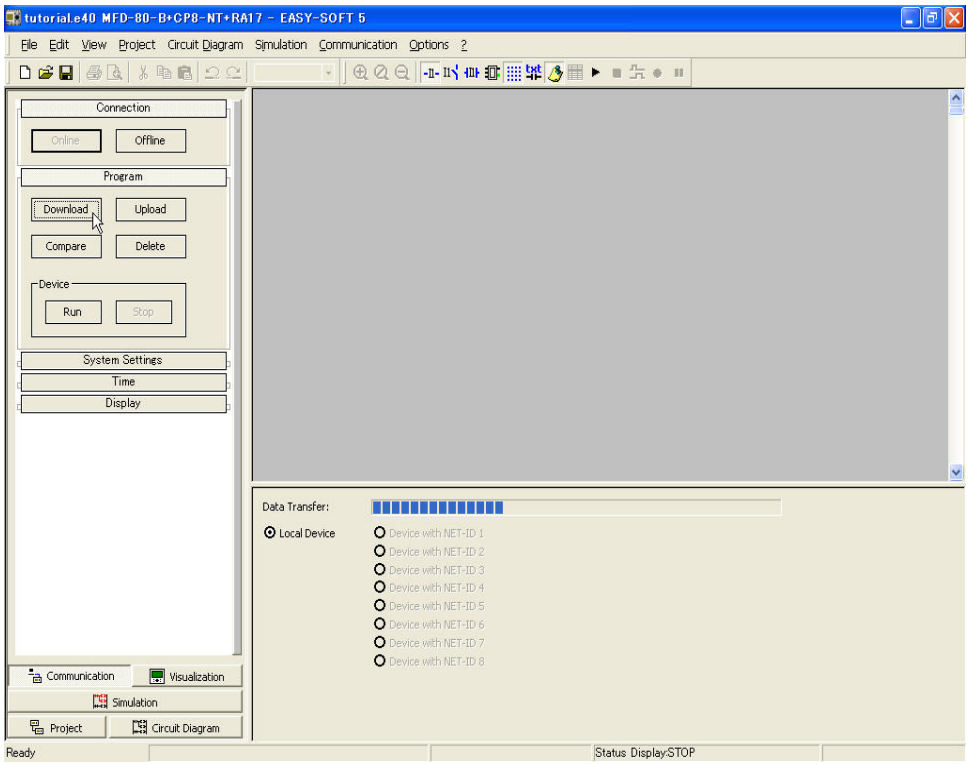
ツールボックス上部の「Connection」の「Online」ボタンを押してください。



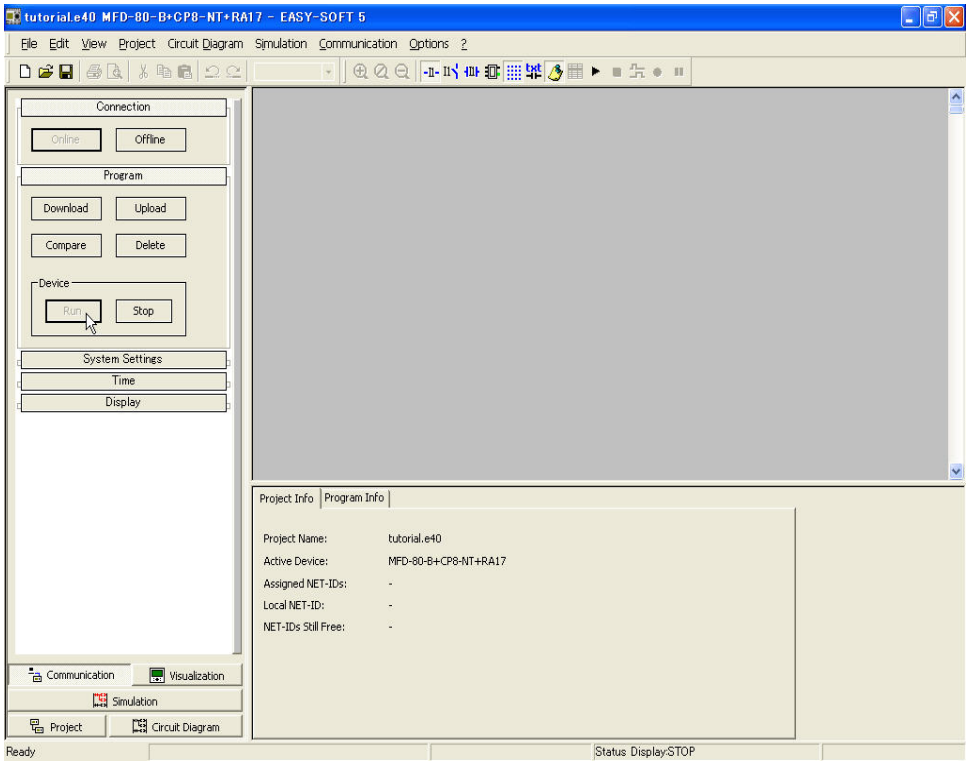
MFD と接続されました。



次に「Program」ボタンを押してください。



「Download」ボタンを押して、プログラムを MFD 本体にダウンロードします。伝送状況は、プロパティフィールドの Data Transfer に表示されます。

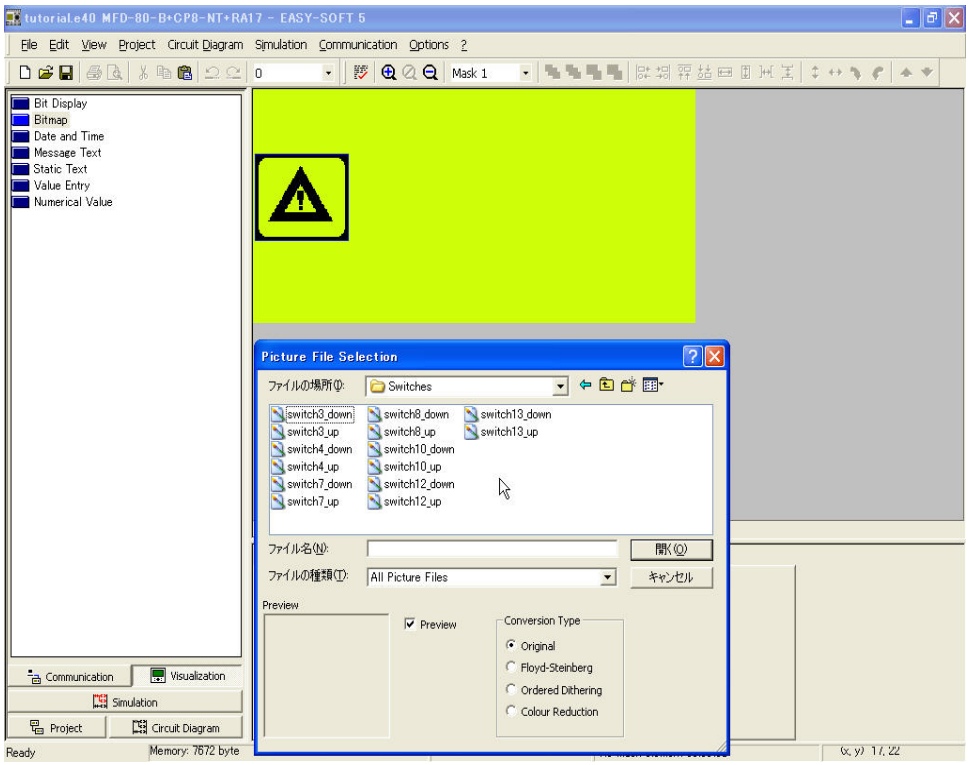


最後に Device 枠の「Run」を押して、MFD 本体を RUN モードにします。
これでプログラムが実行できます。

５．スイッチのアニメーション化

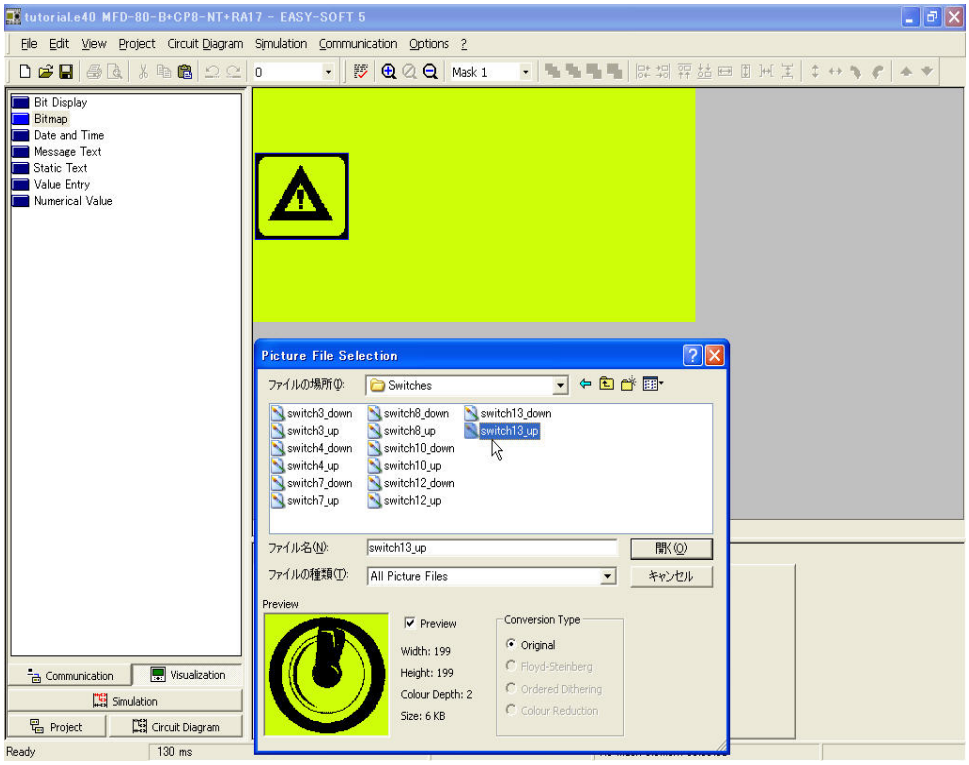
ここでは、以下のようなプログラムを作ってみましょう。

［タスク定義］ I2 をオンにした時、画面には上向きスイッチのビットマップ画像が現われ、オフの時は下向きスイッチのビットマップ画像が現われる。

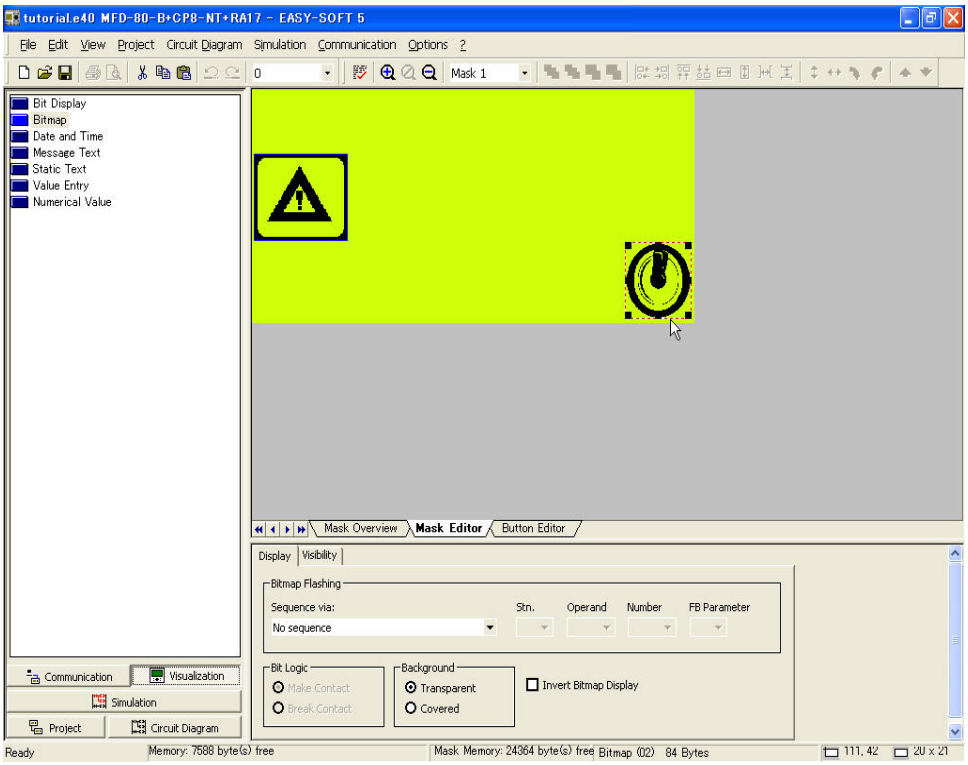


ビットマップ画像を重ねることによって、アニメーションを作ります。

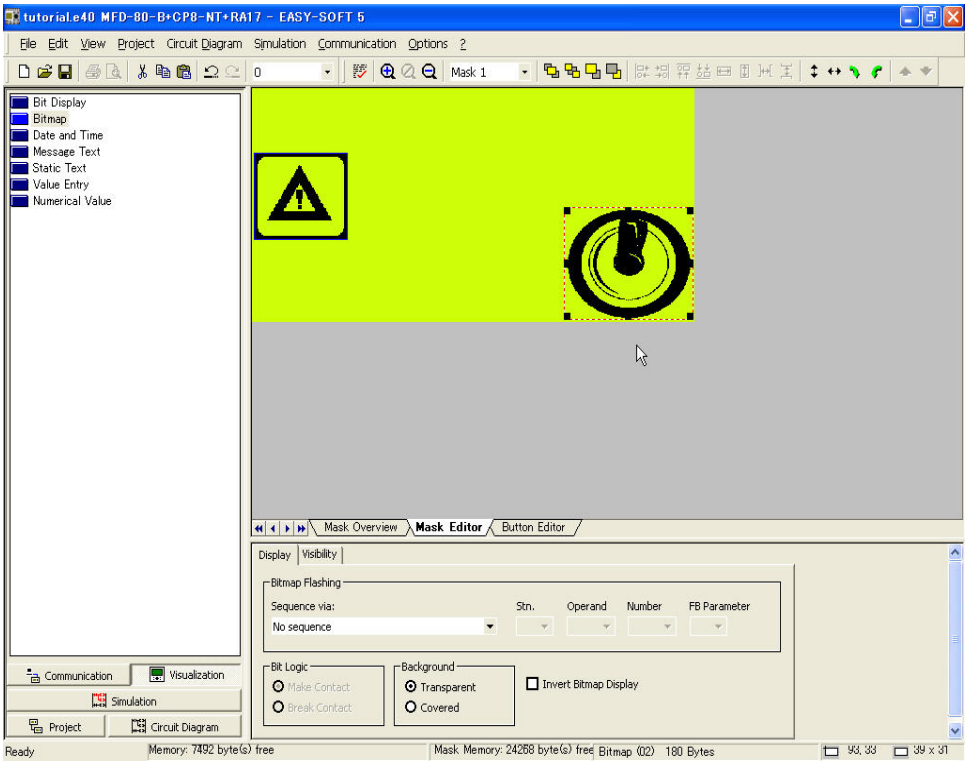
4節のはじめと同様に、Bitmap をツールボックスからワークベンチへドラッグします。



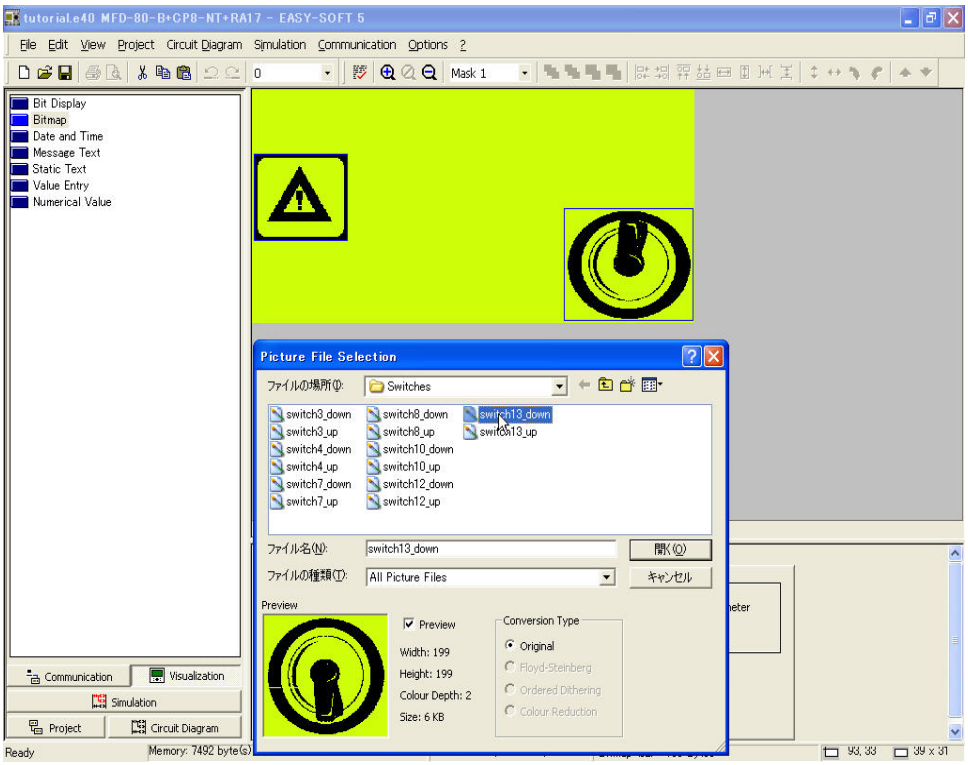
「Switch」フォルダから「switch13_up」を選択し、「開く」。



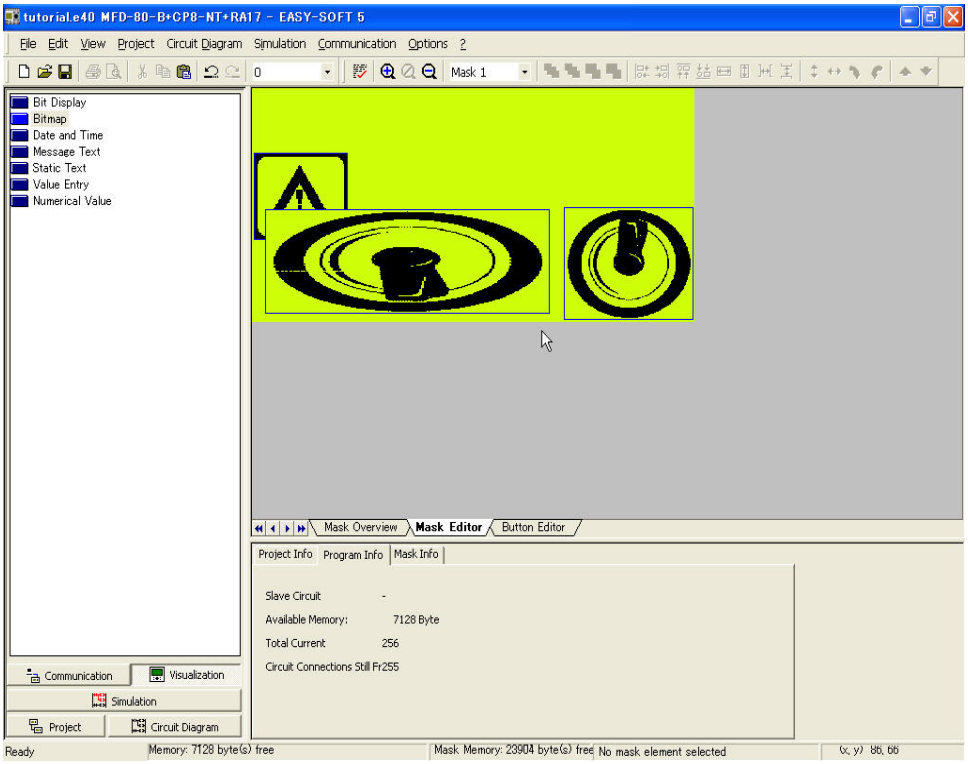
挿入したビットマップを右下に配置しましょう。



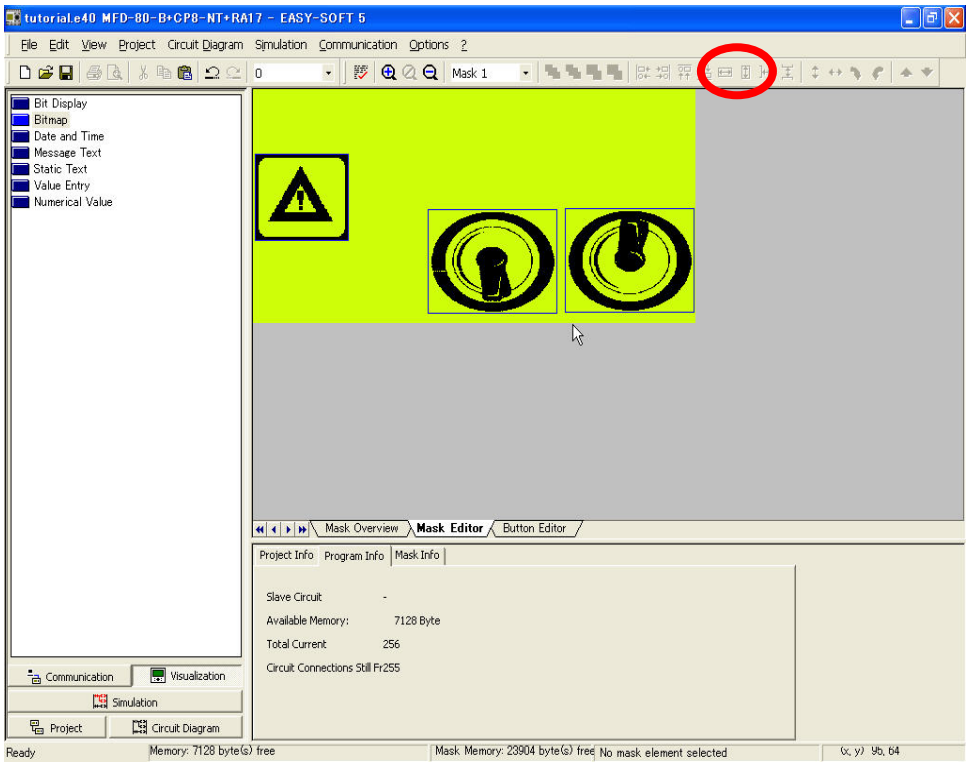
4隅か辺の中央をポイントして 2 方向の矢印にし、大きさを適宜調整します。



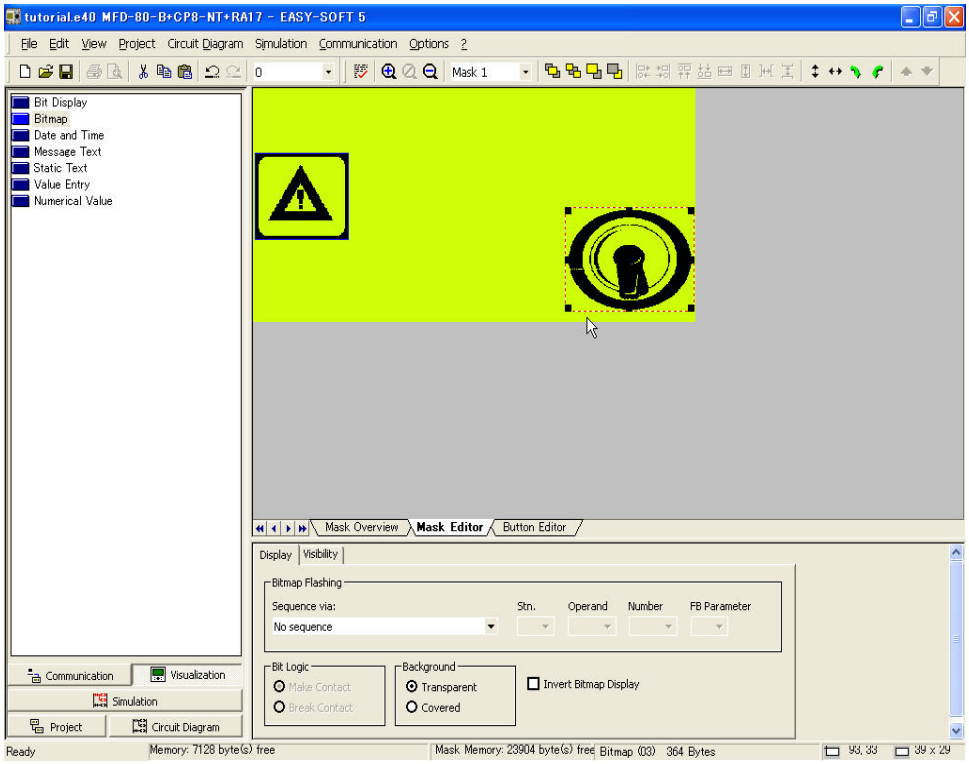
同様に「switch13_down」ビットマップを挿入します。



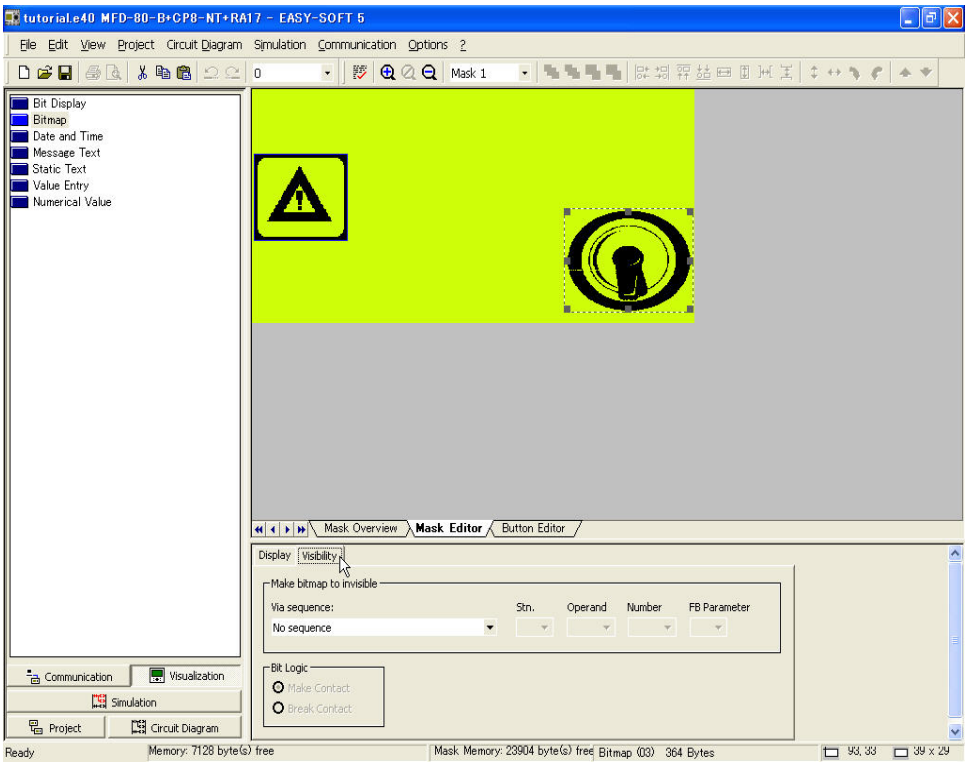
マスク上で適当な場所に置きます。



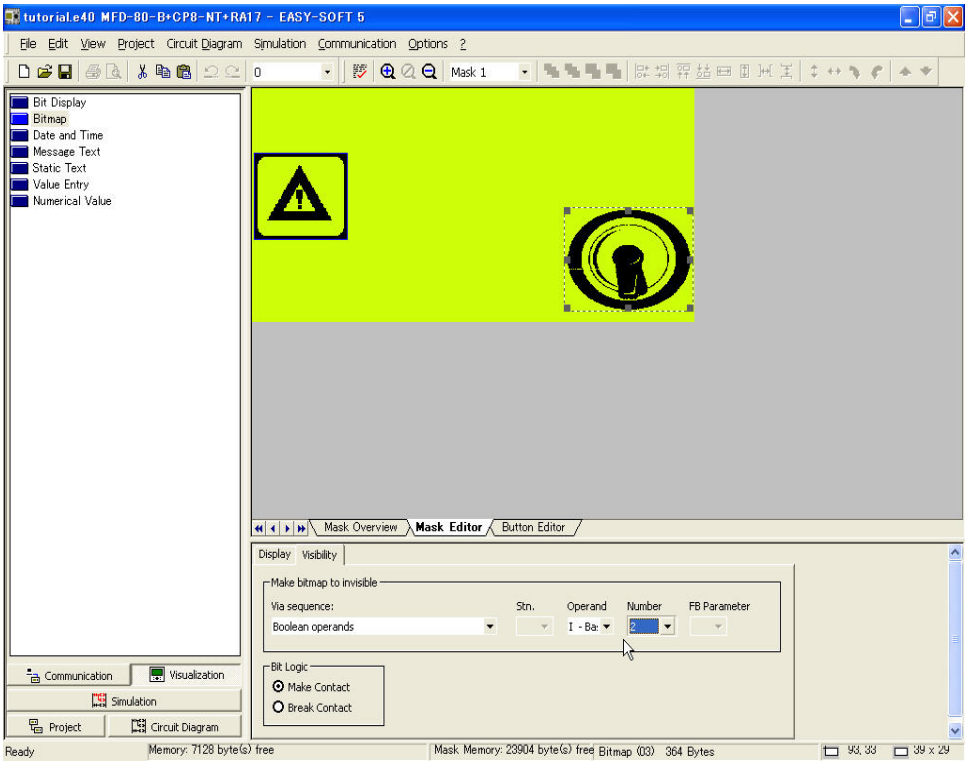
2 つのビットマップの大きさを揃えます。
まず、どちらか一方のビットマップを選択し、シフト OK 押しながらもう片方を選択。
すると最初に選択したものが赤い破線、後で選択したものが黒い破線で囲まれます。
この状態になると  Equal Height（高さを合わせる）ボタンと、  Equal Width（幅を合わせる）ボタンが有効になりますので、それぞれ 1 回ずつクリックすると、2 つの画像のサイズがぴったり同じになります。



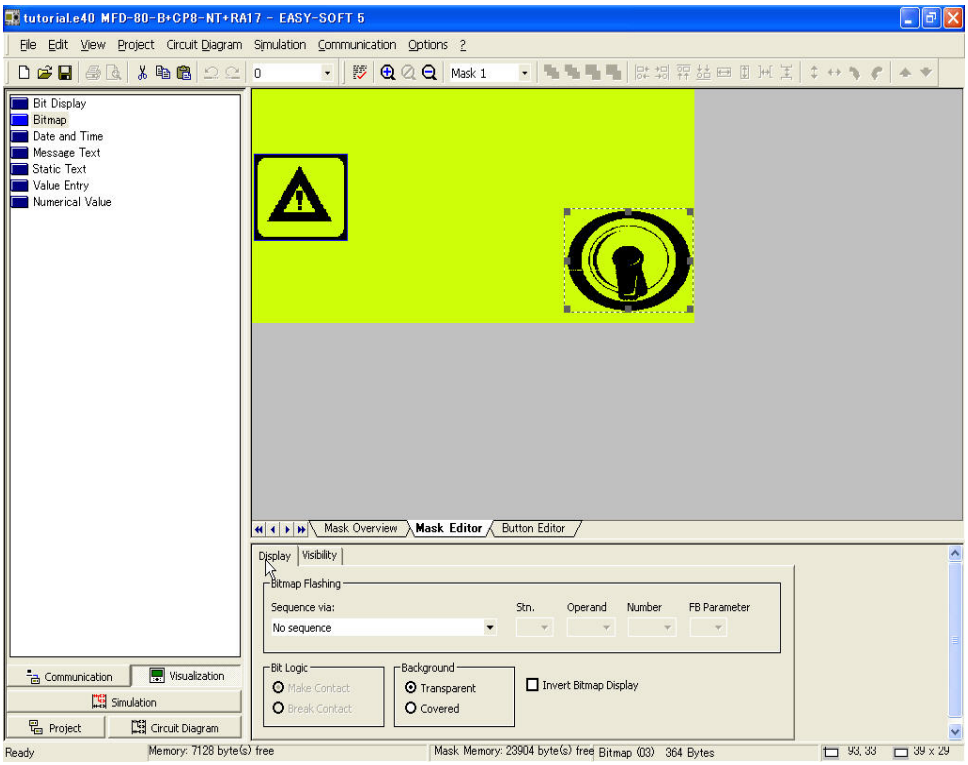
マスク上のビットマップの無い部分をクリックし、選択を解除、再び「switch13_down」のみを選択し、ドラッグで「switch13_up」の上にぴたっと重ねます。



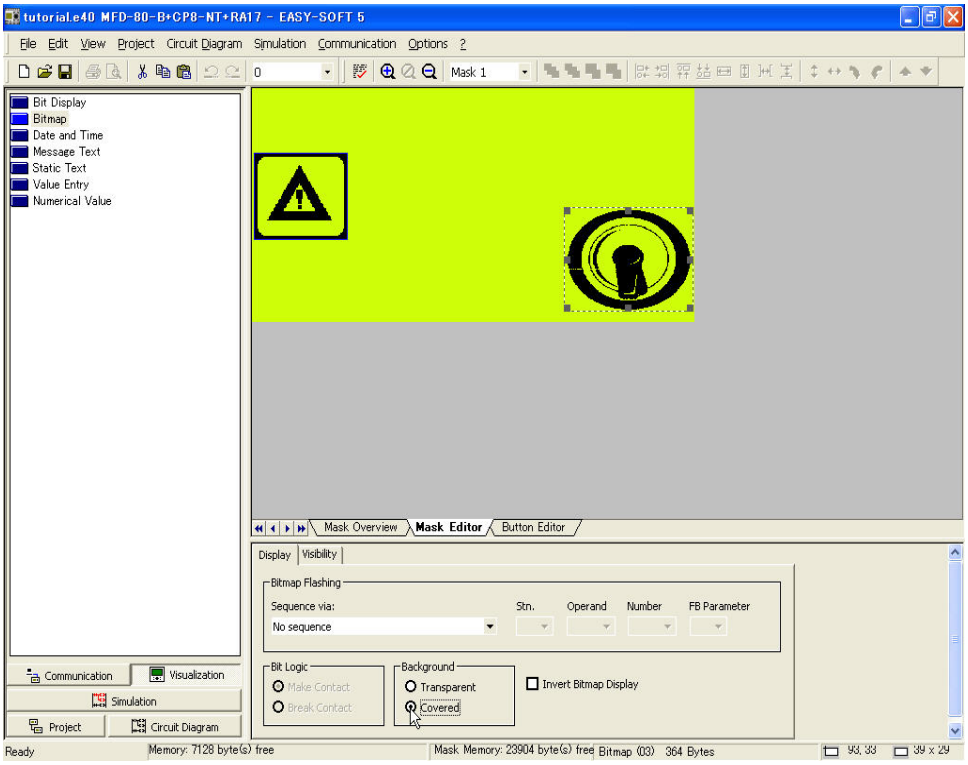
ここで、I2 がオンの時、下のビットマップ（上向きスイッチ）が現われて、上のビットマップ（下向きスイッチ）が消えるように設定します。現在の状態で上のビットマップ（下向きスイッチ）が選択されている状態ですので、ここで「Visibility」タブをクリック。



「Make Bitmap to invisible（ビットマップを不可視にする）」枠内で、「Via sequence」を「Boolean operands」にします。Operands 欄では「I-Basic Unit Input」を選択。Number 欄では「2」を選択。Bit Logic では「Make Contact」を選ぶと、I2 がオンの時、上のビットマップ（下向きスイッチ）が消えるという設定になります。

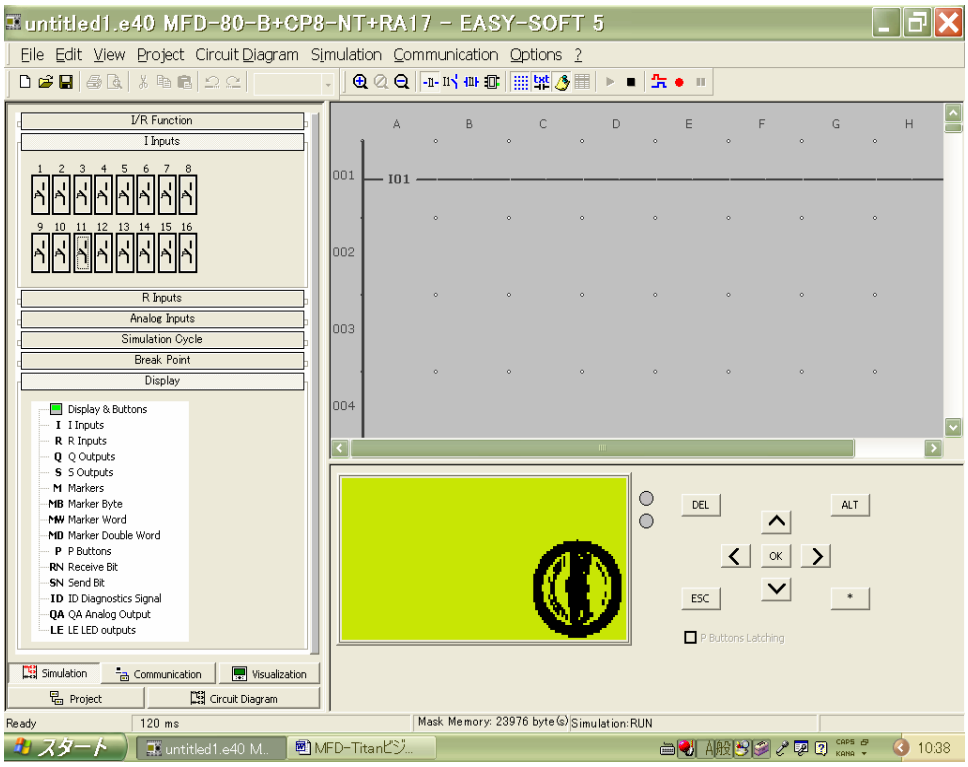


しかし、このままだと、I2 がオフの状態で下のビットマップ（上向きスイッチ）が透けて見えてしまう状態です。

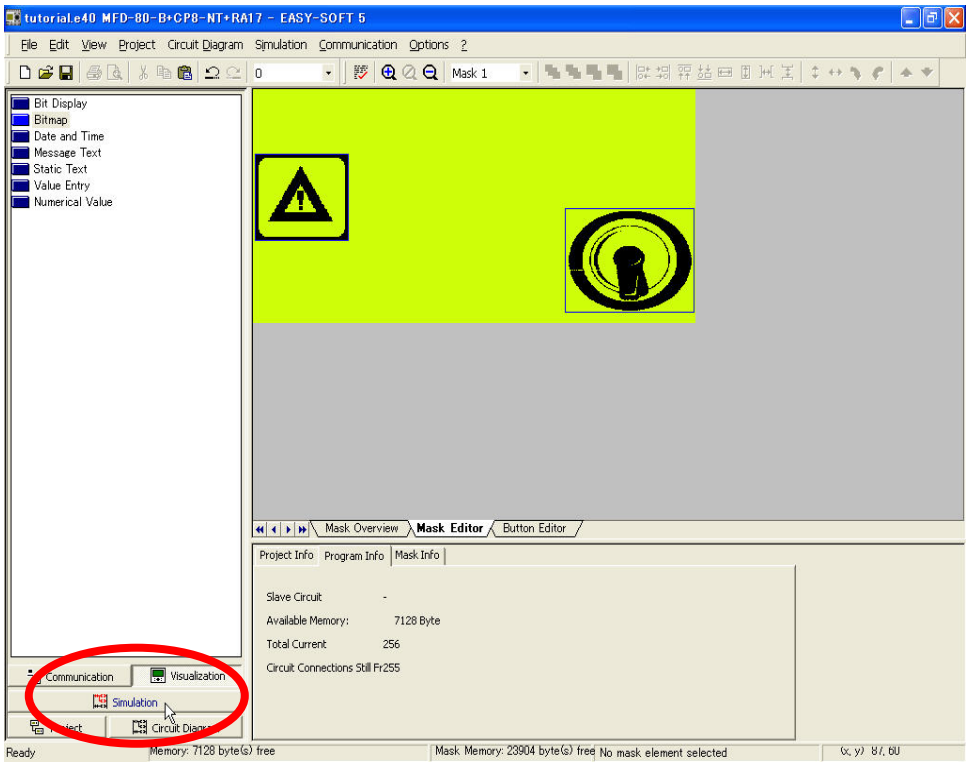


そこで「Display」タブをクリックし、Background 枠の Transparent（透明）を Covered（不透明）にします。これは上のビットマップの背景が下のビットマップを隠すという設定です。アニメーション作成の際には必要になってきますので、必ず行ってください。

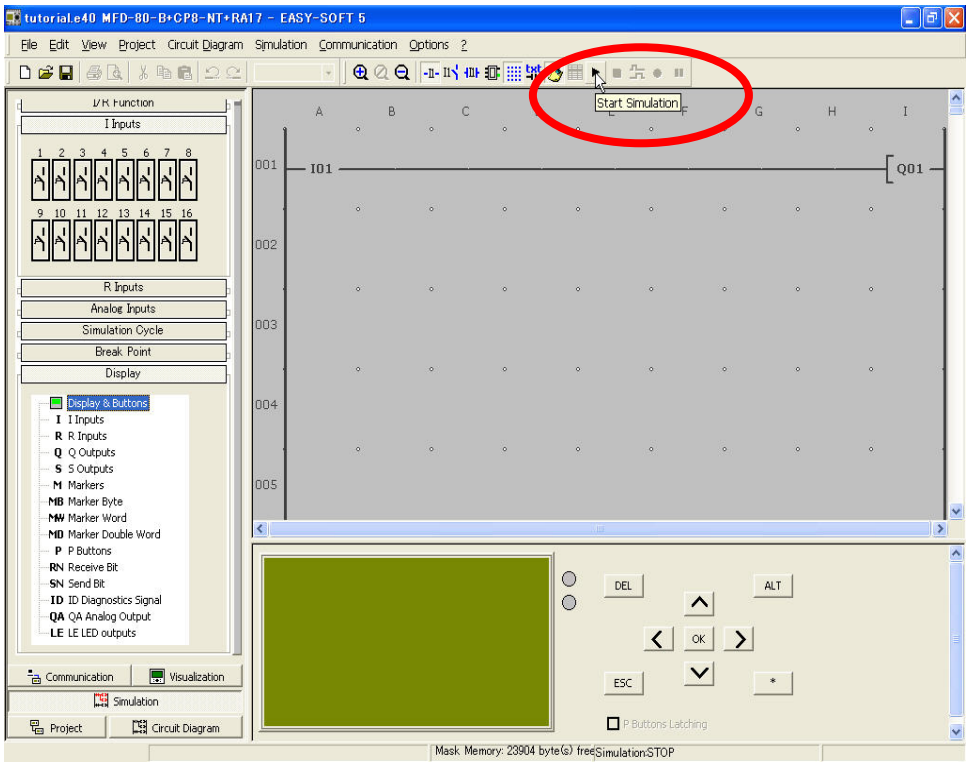
もし個の設定をしないと、次ページ上図のシミュレーションように、2 つのビットマップが透けて同時に見えてしまいます。



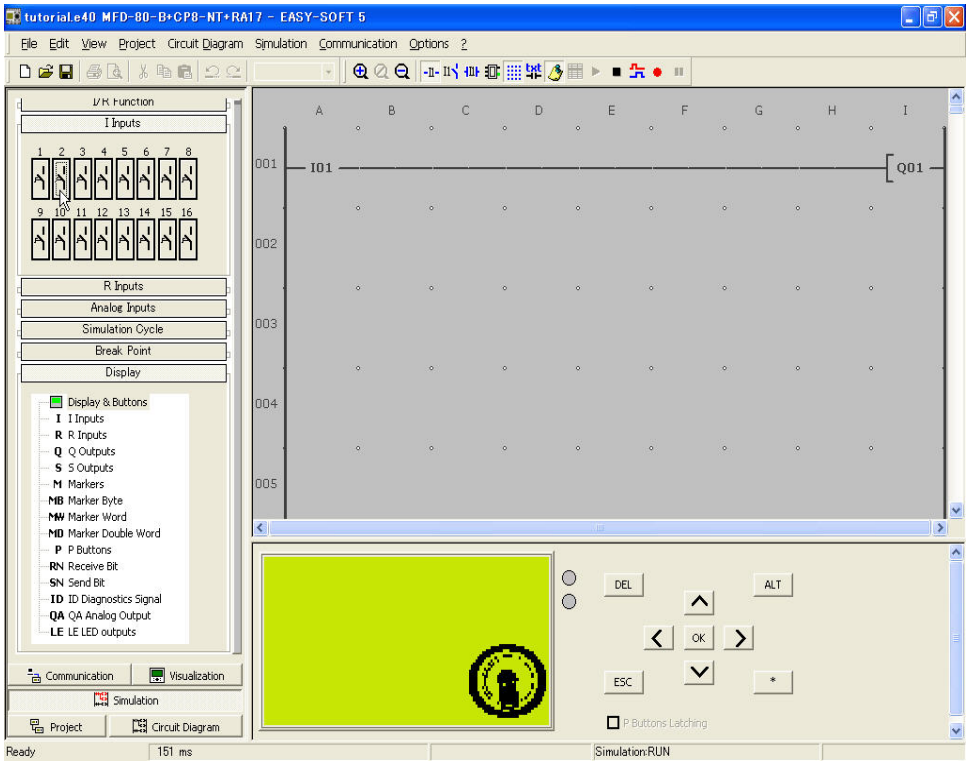
失敗例：上のビットマップの Background（背景）を Covered（不透明）にしなかった例。



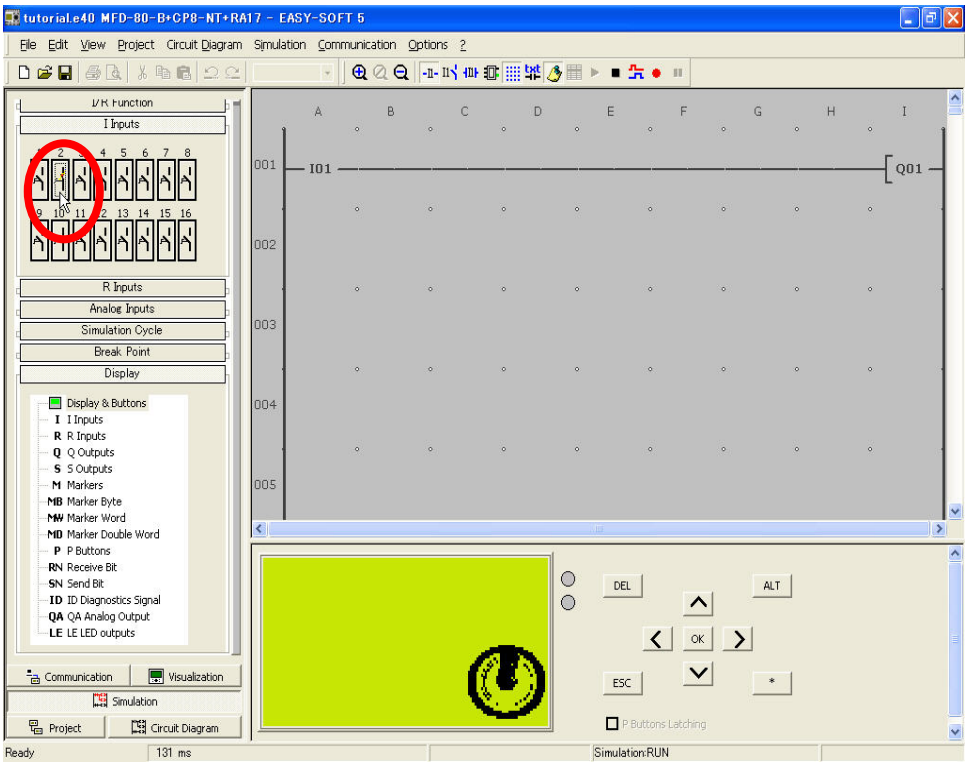
ここで、マスク動作のシミュレーションをしてみましょう。「Simulation」ボタンをクリックします。



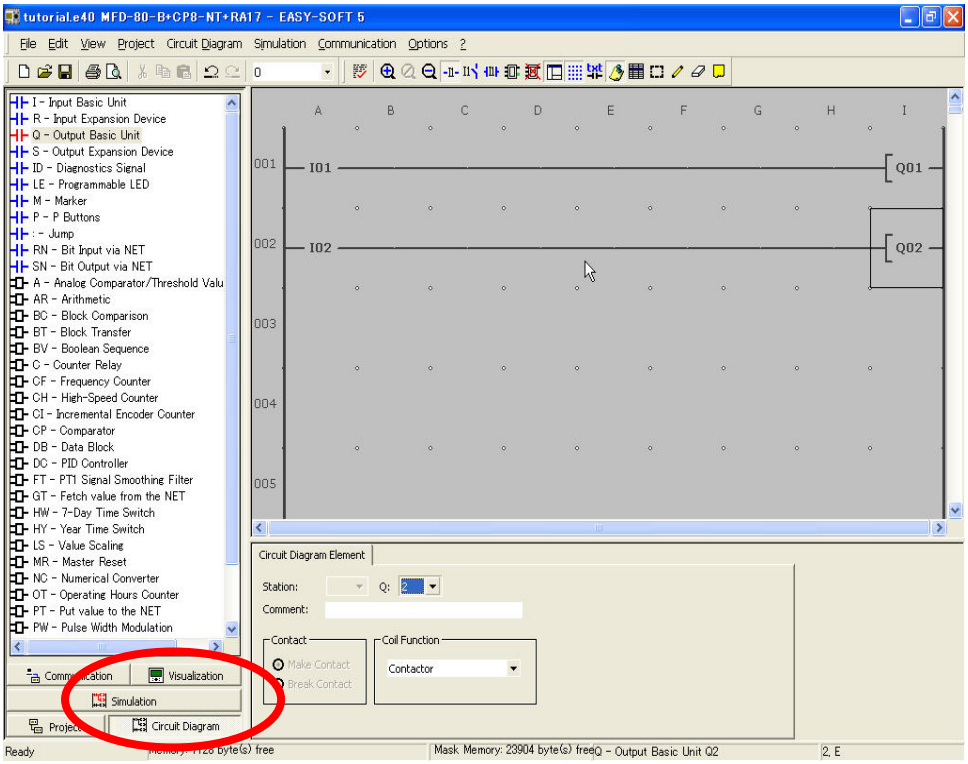
前節と同じように「Input」ボタン押して各入力接点を表示させ、また「Display」ボタンを押し、「Display&Button」を押して、プロパティフィールドにMFDの画面とボタンのシミュレーション画面を表示させます。「Start　Simulation」ボタン ▶ を押します。



現在 I2 がオフ状態なので、I2 をクリックして投入してみてください。



画面上で上向きスイッチ（下のビットマップ）が現われました。



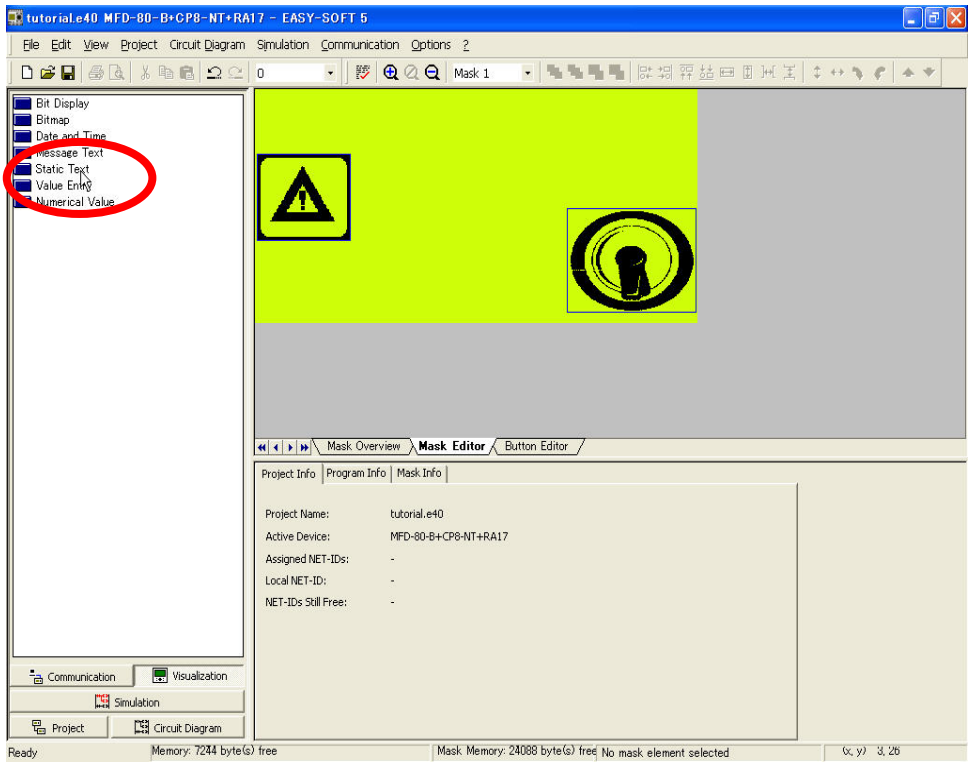
次に、プログラムに必要なもう一つの部分であるラダー回路を作成します。上図のような簡単な回路になりますので、ツールボックスの「Circuit　Diagram」をクリックし、必要な回路要素をドラッグして作成してください。今回は入力・出力ともパラメータは2ですので、プロパティフィールドで設定するのを忘れないでください。

17 ページからの MFD 本体との通信確立方法、ダウンロード方法を参考に、プログラムをダウンロードして、実行してみてください。

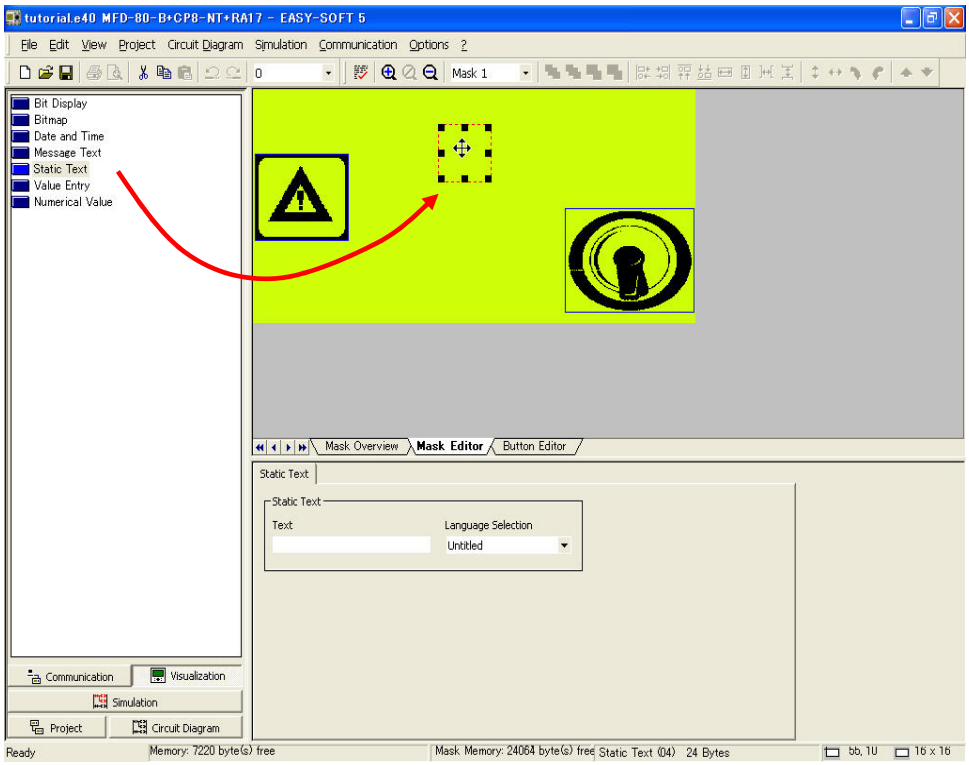
６．キーパッドからの入力画面作成

ここでは以下のようなプログラムを作ってみましょう。

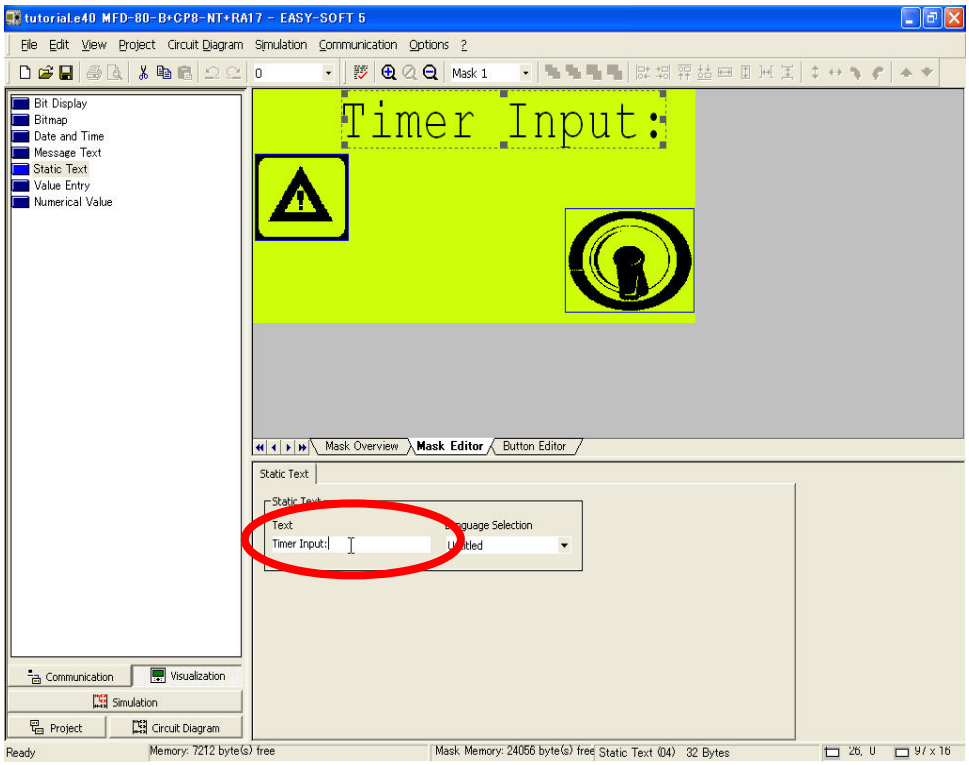
[タスク定義] MFD のキーパッドからタイマの設定値を入力することができるような。タイマを設定します。I3 がトリガーとなって、その設定値にしたがってタイマが動作し、タイムアウトした際は出力 Q3 がオンになるようにします。



まず「タイマ入力」という表示を作りましょう。

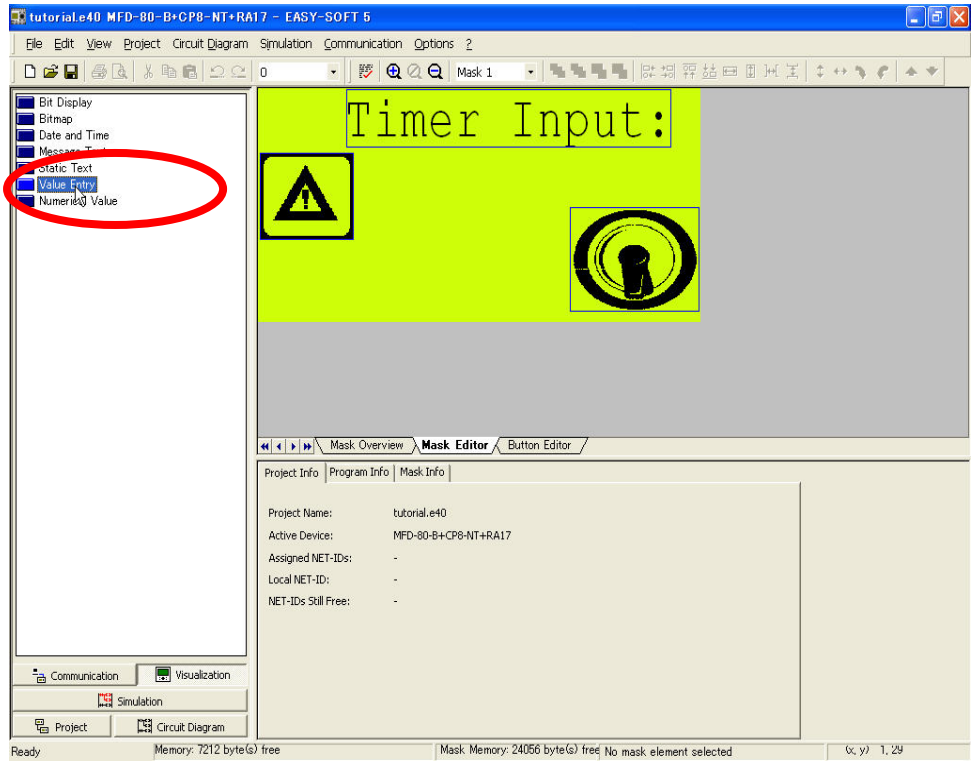


ツールの「Static Text（静止文字）」をワークベンチへドラッグします。表示用の静止文字入力枠ができます。

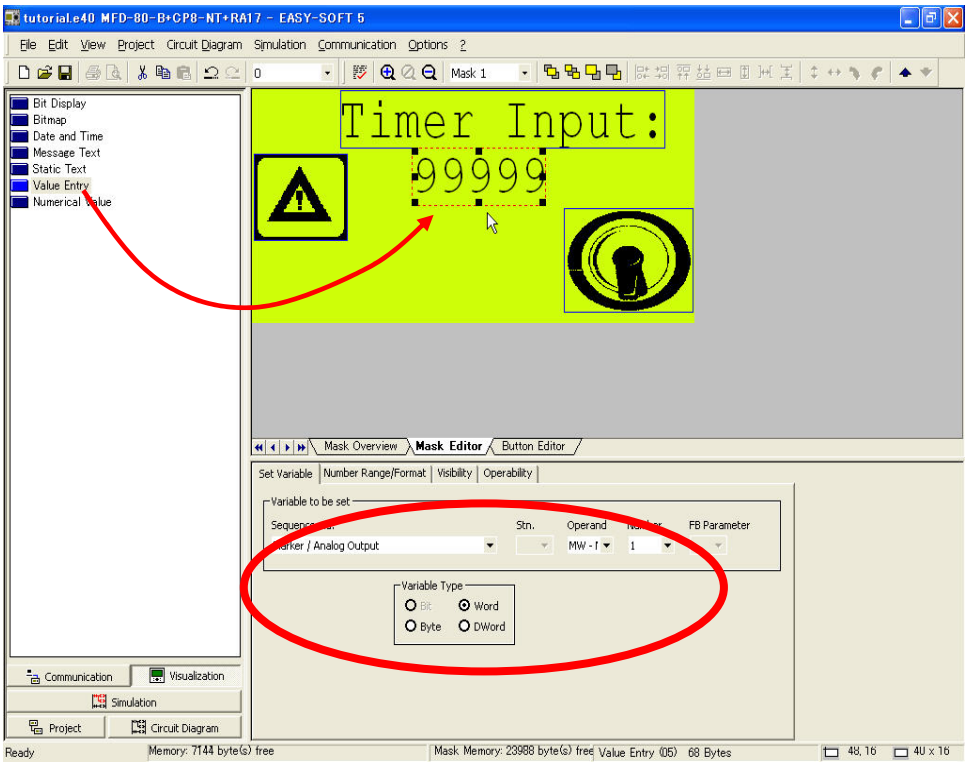


プロパティフィールドの「Text」で、”Timer Input :”と入力してください。

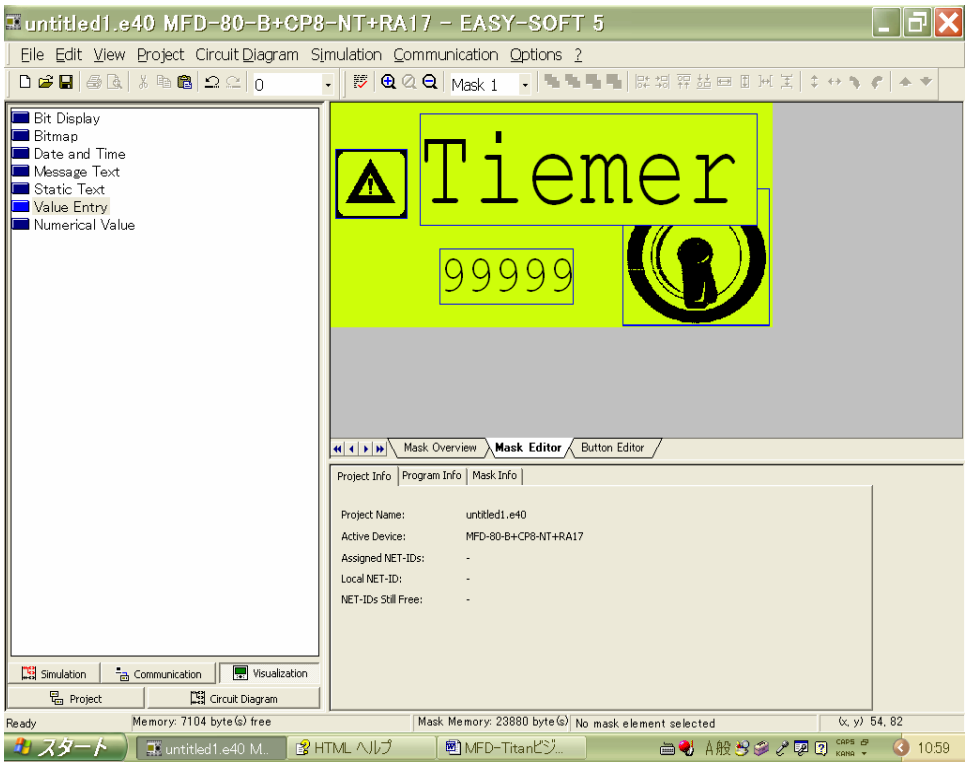
参考:ビジュアル画面でこのように入力できる言語は、現在のところ、英語をはじめとするヨーロッパ諸国言語 10 言語に限られています。ただし、**ビットマップ画像として、漢字を含む日本語を表示させることもできますので**(８． 日本語のビットマップ作成 52 ページ)をご参照ください。



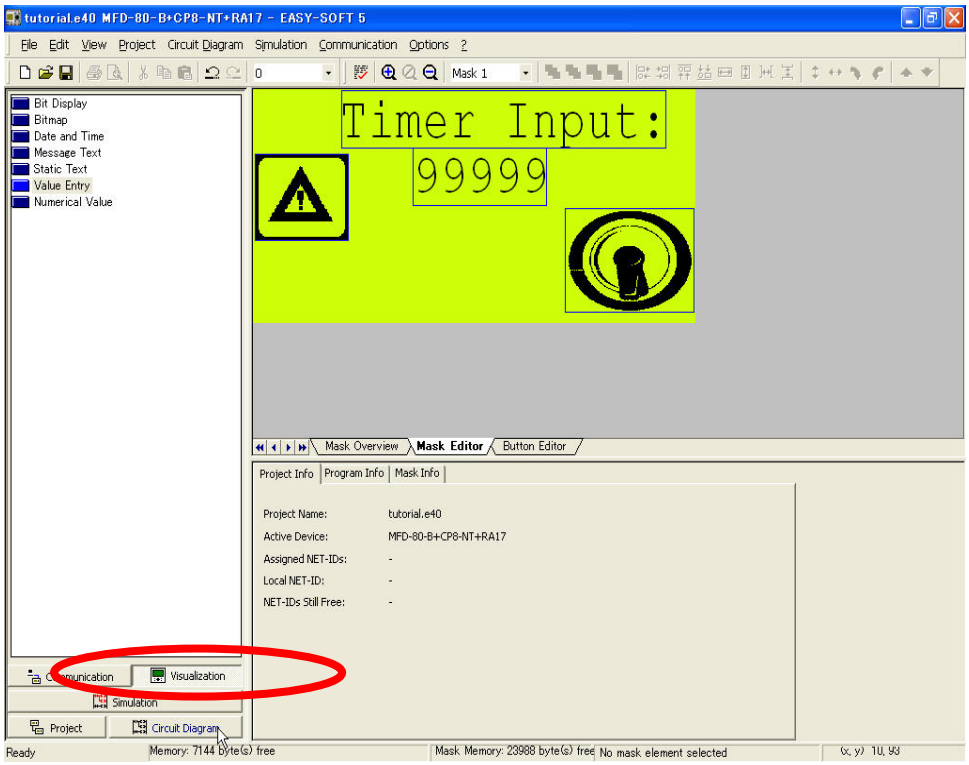
次に実際に入力値を入力する枠を作成します。ツールの「Value Entry」です。



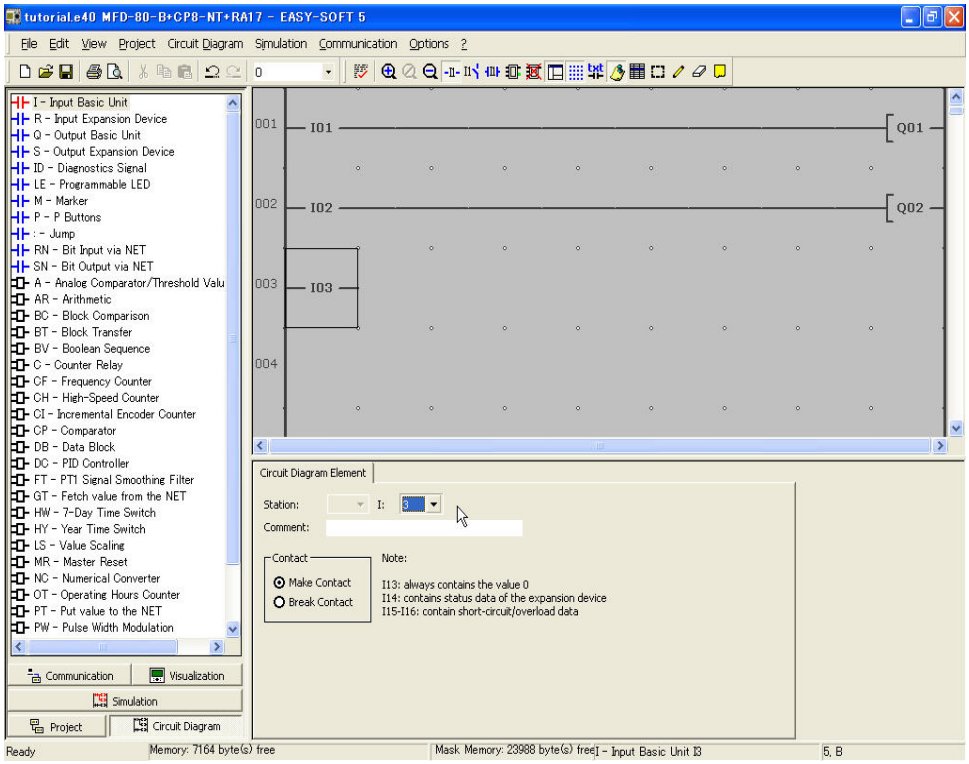
ワークベンチにドラッグして、位置を静止文字の下、中央へ持ってきます。
また、Value Entry 枠が選択されている状態で Sequence Via を「Marker/Analog Output」とし、Operands で MW-1（マーカワード）を選ぶか、Variable Type で Word を選択してください。参考：マーカについての詳細は、「MFD-Titan 多機能ディスプレイ スタートアップガイド ラダープログラム編 Vol.2」63 ページをご覧ください。



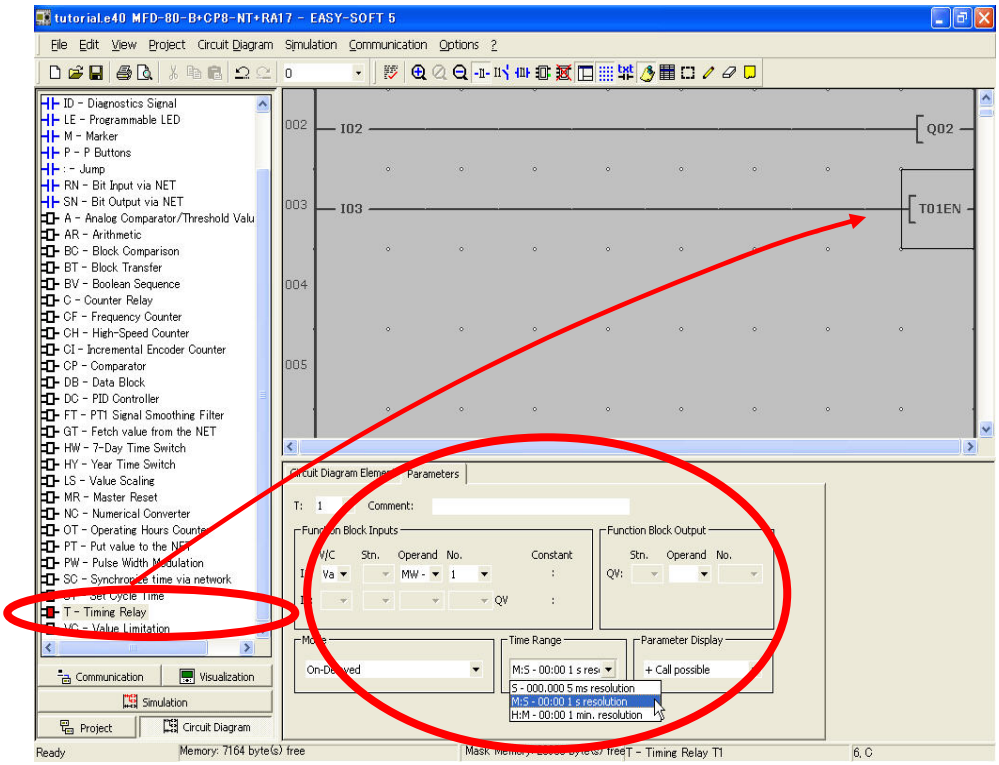
参考**文字の大きさについて**：文字のサイズは標準サイズと倍角サイズの 2 種類があります。Static Text の領域枠を縦に伸ばすと文字は自動的に倍角文字になります。



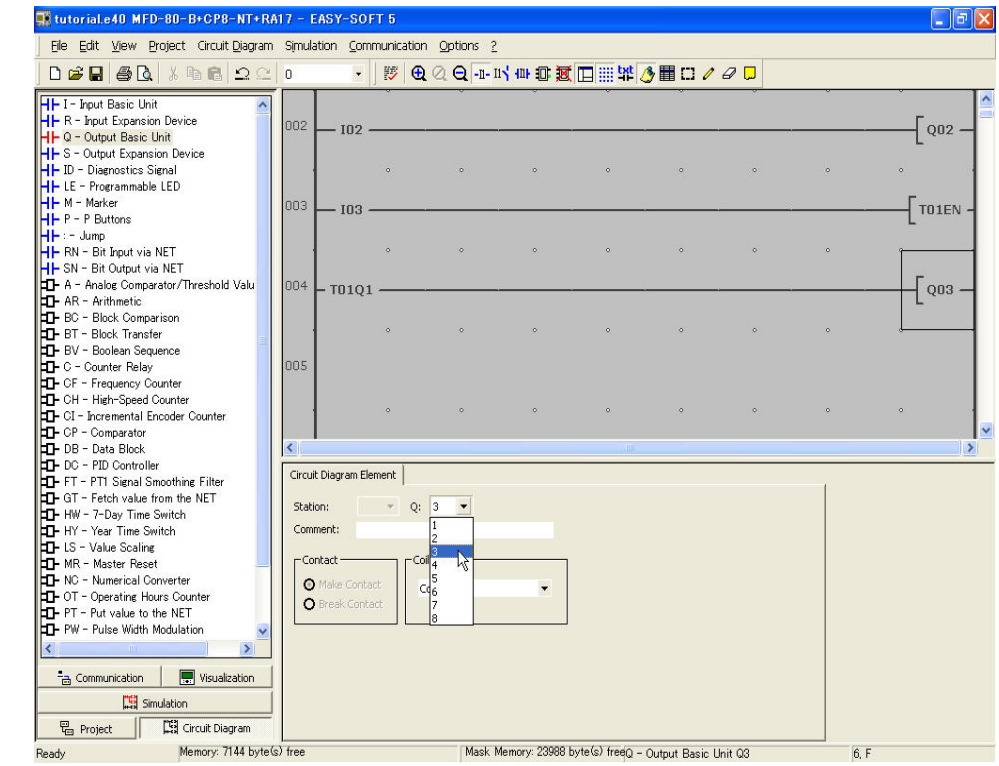
次にラダー回路を作成しますので、「Circuit　Diagram」をクリックしてください。



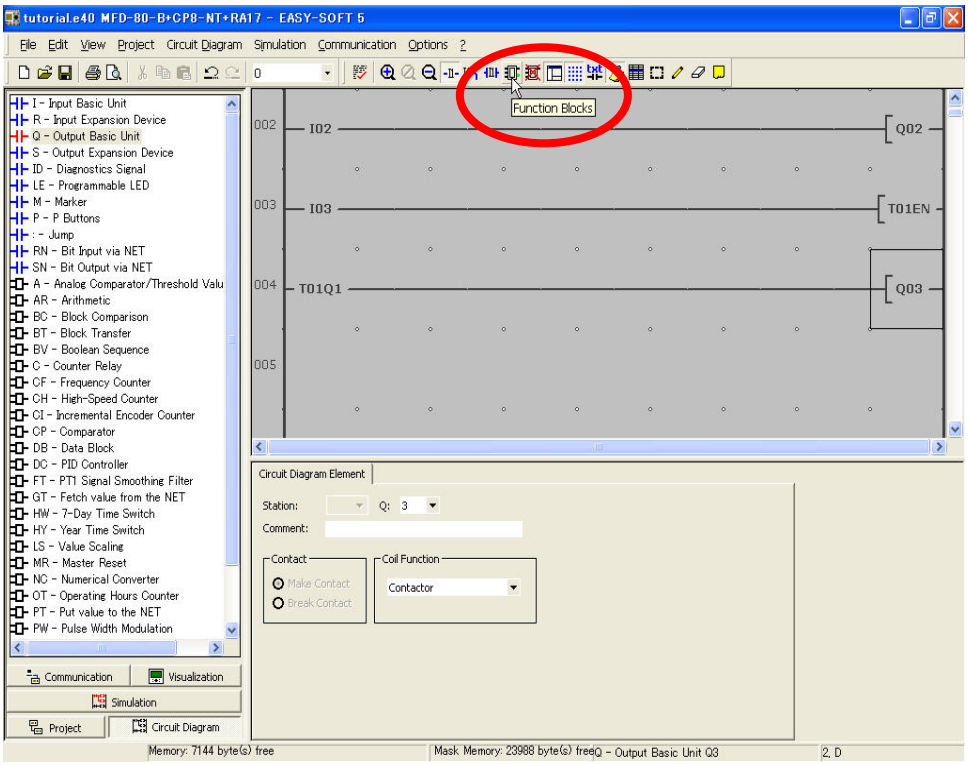
I を 3 列目の左端に置き、パラメータを 3 と設定。



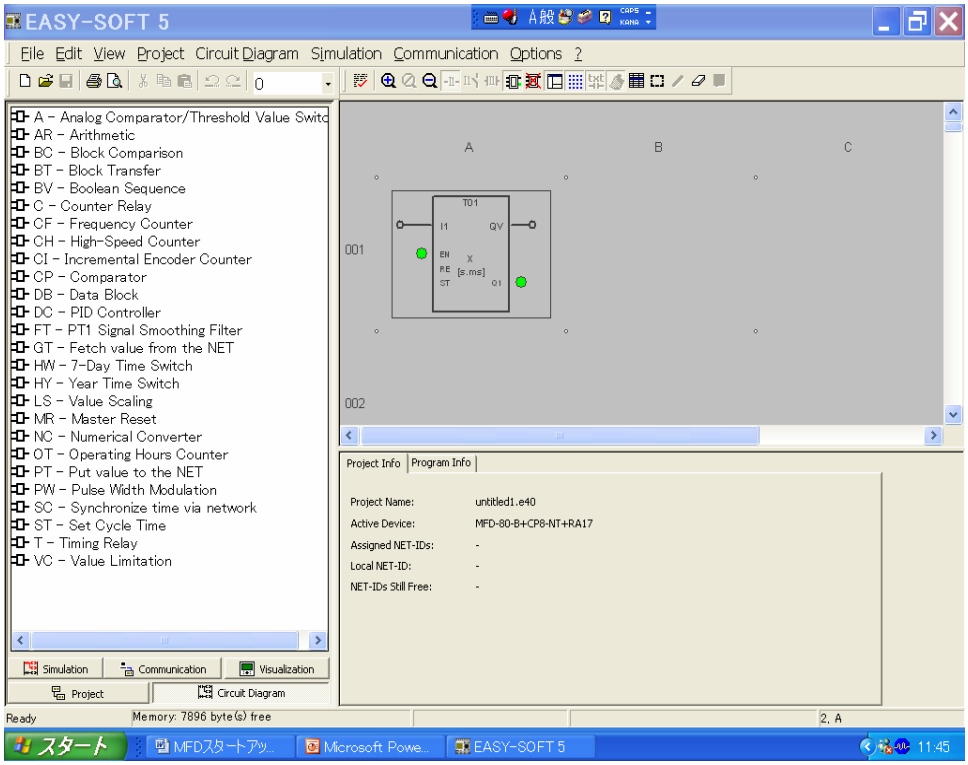
ツール「T-Timing Relay」を 3 列目の右端へドラッグ。これがタイマトリガーになります。
Function Block Inputs で V/C を「Variable」 Operands を「MW-Marker Word」 No.を「1」にします。また、Time Range で秒単位（M:S-00:00 1 resolution）を選択します。Mode は On-Delayed のままで結構です。



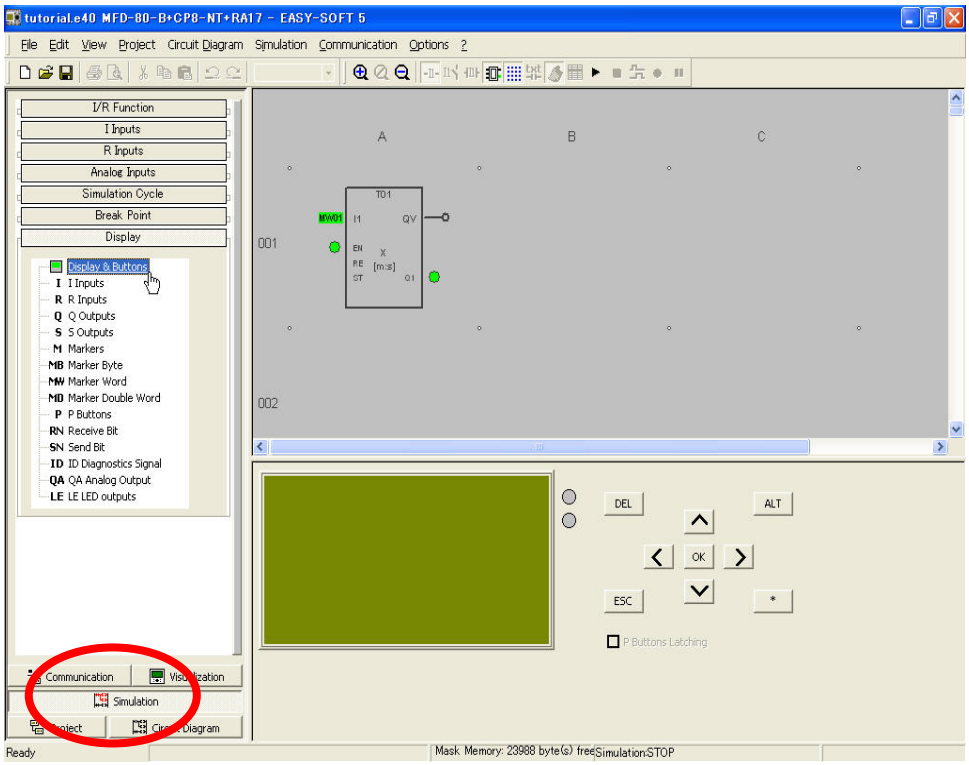
4 列目左端にタイマ出力 T01Q1 を配置し、右端に Q3 を挿入します。これでタイムアウトした際には Q3 にシグナル出力が出るようにプログラムされました。



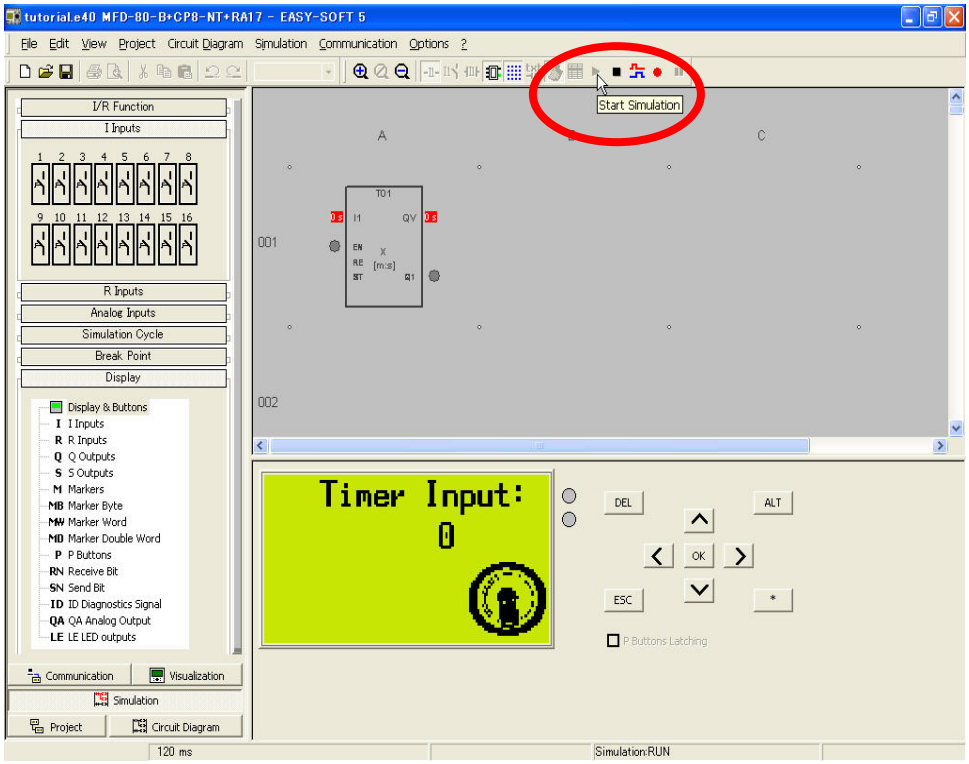
シミュレーションに入る前に、タイマの動作を見やすいように、Function Block ビューに切り替えておきましょう。



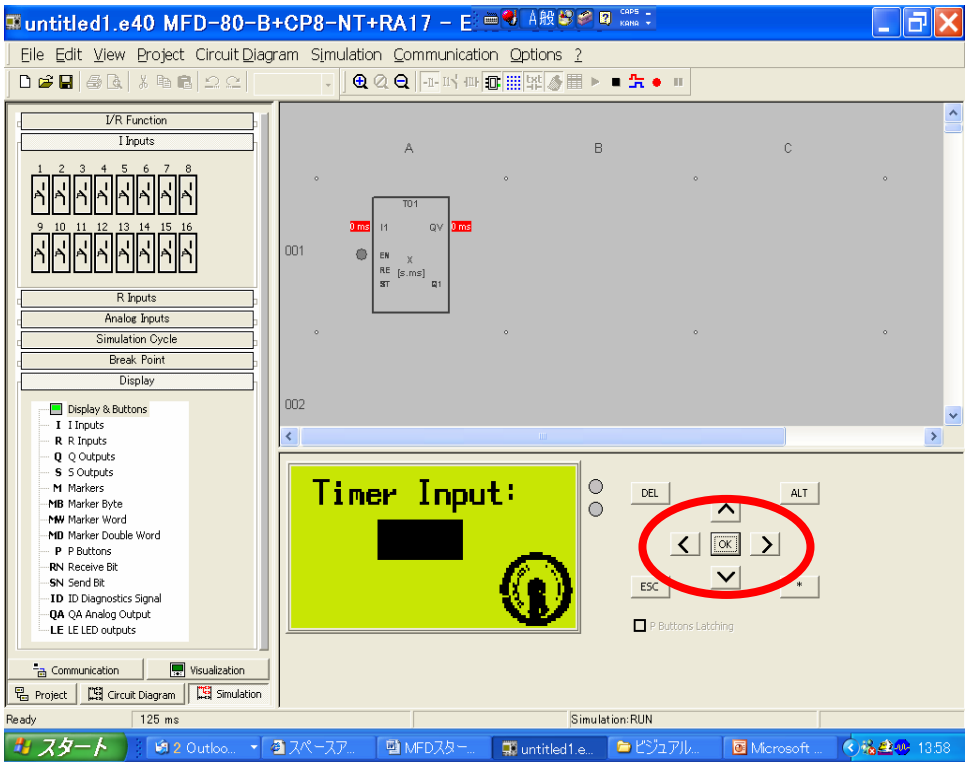
上のような画面になります。



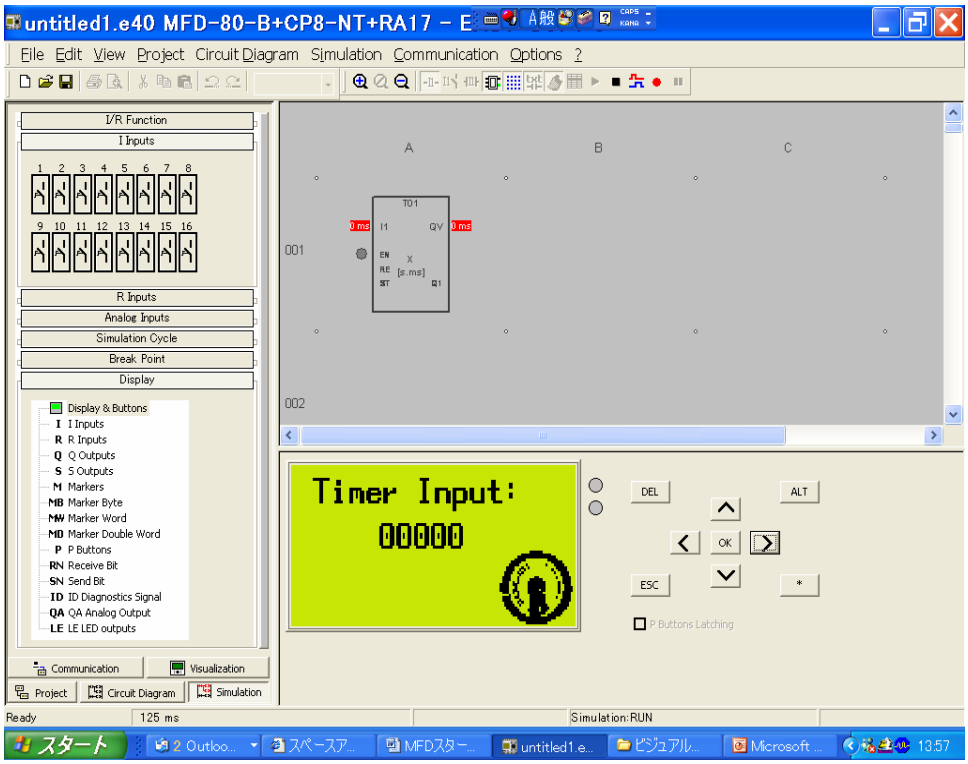
ツールボックス内で「Simulation」ボタン、「Input」ボタン、「Display」ボタン→「Display＆Button」ボタンを押します。プロパティフィールドに画面とボタンのダミーが表示されます。



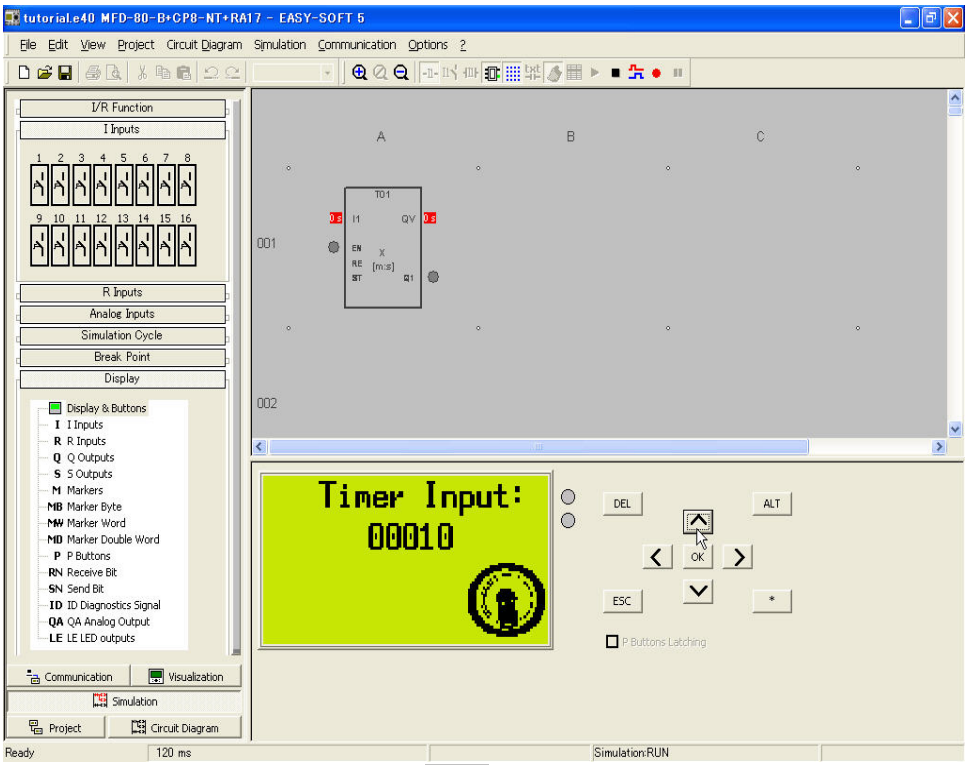
シミュレーション実行ボタン（Start　Simulation）を押すと上のような画面になります。



ダミーボタンの OK ボタンを 1 回押すと、入力枠内で黒のカーソルが点滅します。

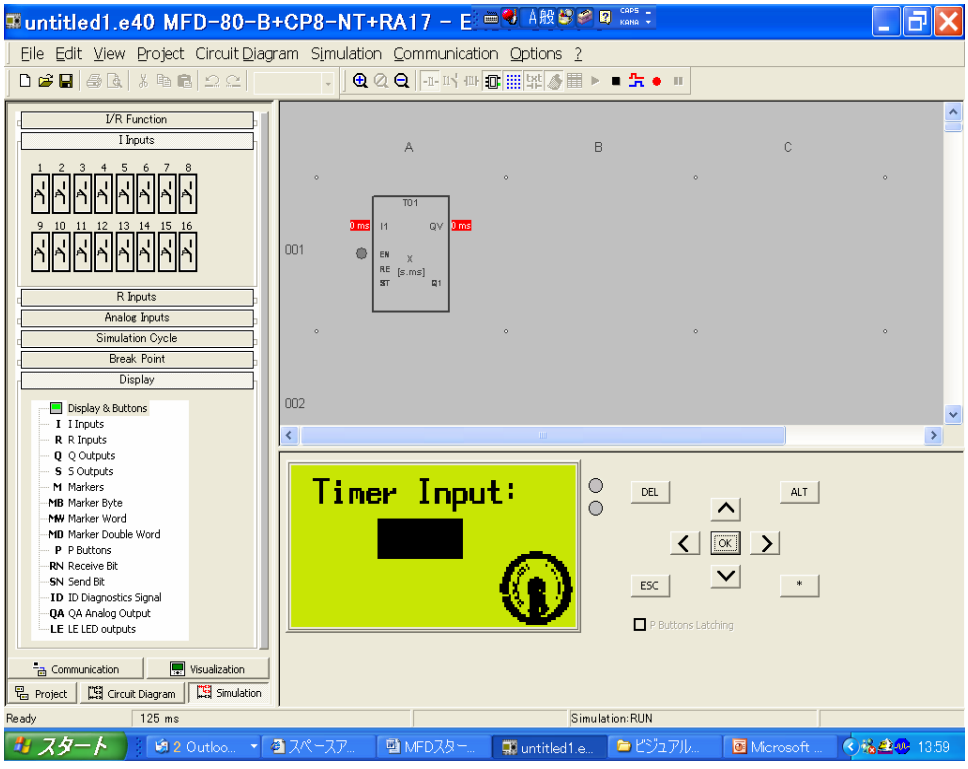


もう 1 回 OK ボタンを押すと、入力枠の全桁数が 0 で表示され、入力できる状態になりました。入力対象になっている桁の 0 が点滅しています。最初は1の位が点滅しています。

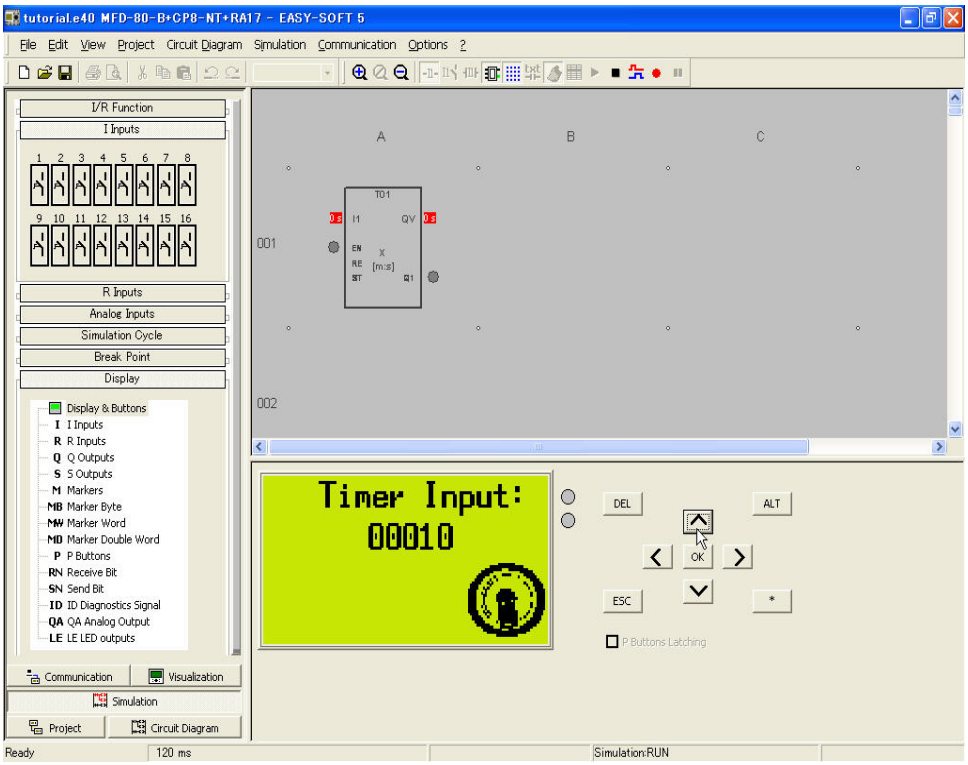


〈 ボタンで 10 の位に移動して、へ ボタンで値を1にします。

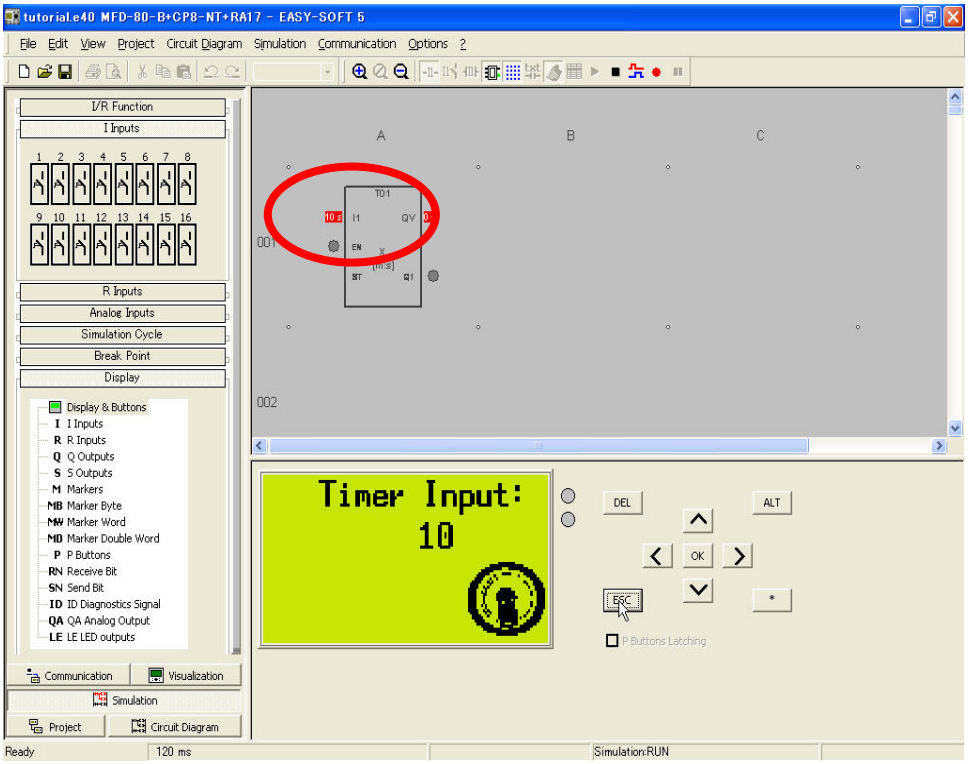
これでタイマ時間が 10 秒に設定できました。



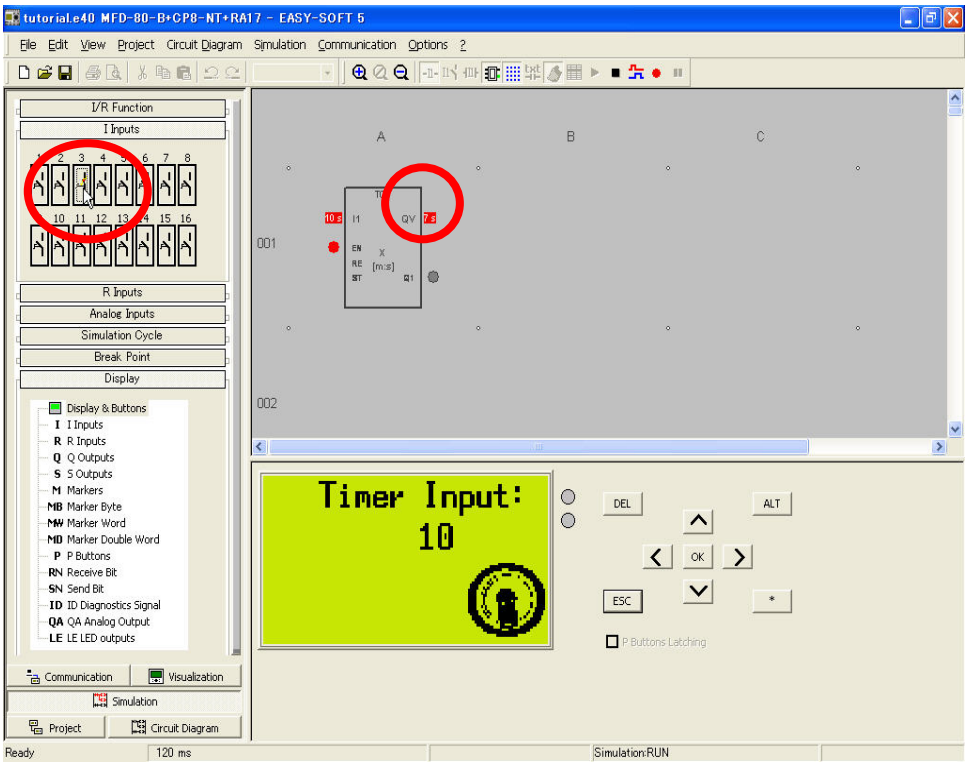
OK ボタンを押して値を受け入れます。再び黒のカーソルが点滅します。



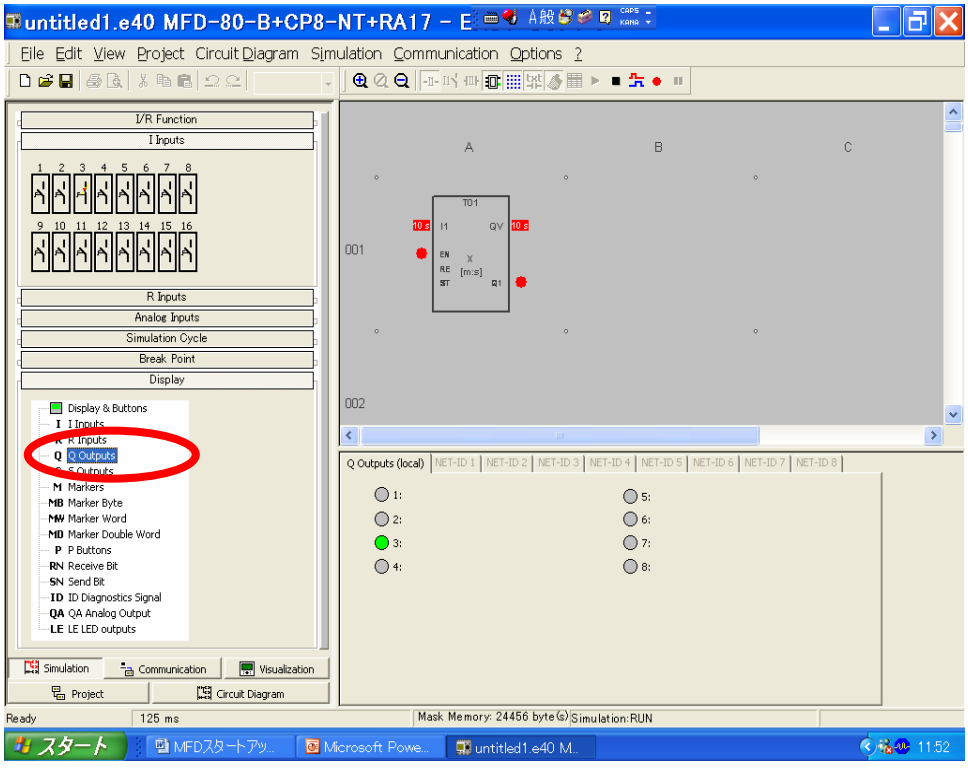
ESC キーで値を確定します。



タイマ 01 の設定値が 10sに設定されたのが確認できます。



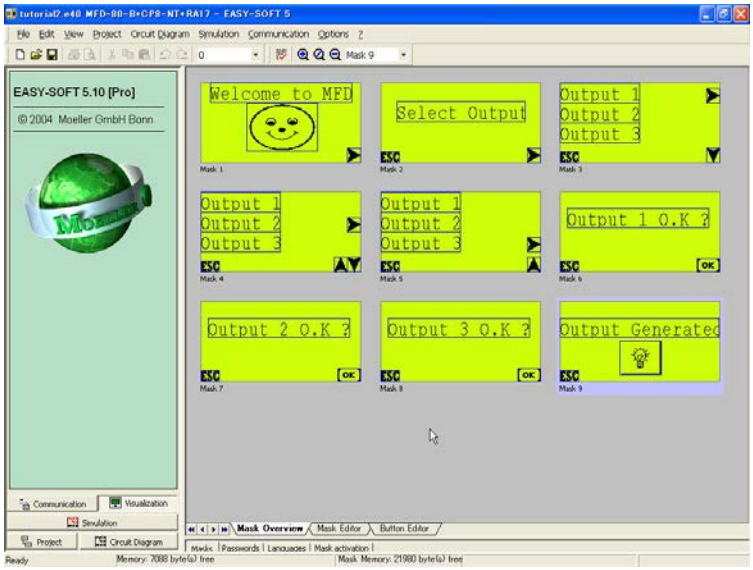
ここで I3 を投入。QV 実効値に注目してください。直ちにカウントを始めています。



タイムアウトしたところで、Display ボタン下の Q Output を押すと、プロパティフィールドに出力の状態が表示されます。Q3 が出力されているのが緑の点灯で確認できます。

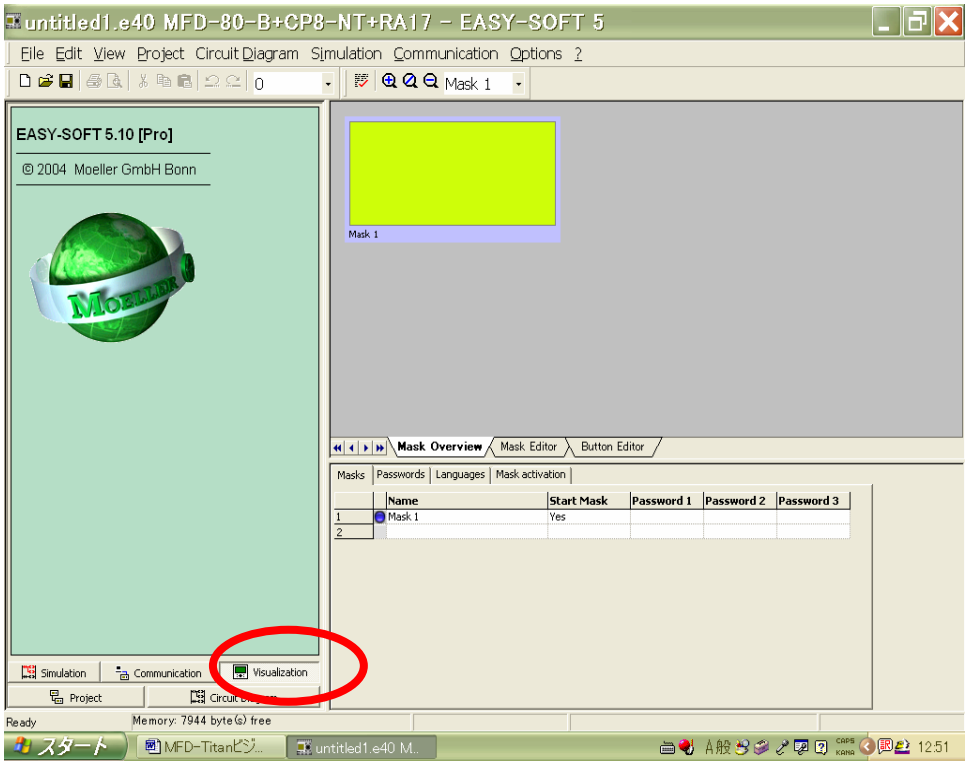
後は、17 ページからの MFD-Titan との通信、プログラムダウンロード方法に従って、プログラムをダウンロード、実行してください。

７．キーパッドで切り替わるメニュー画面

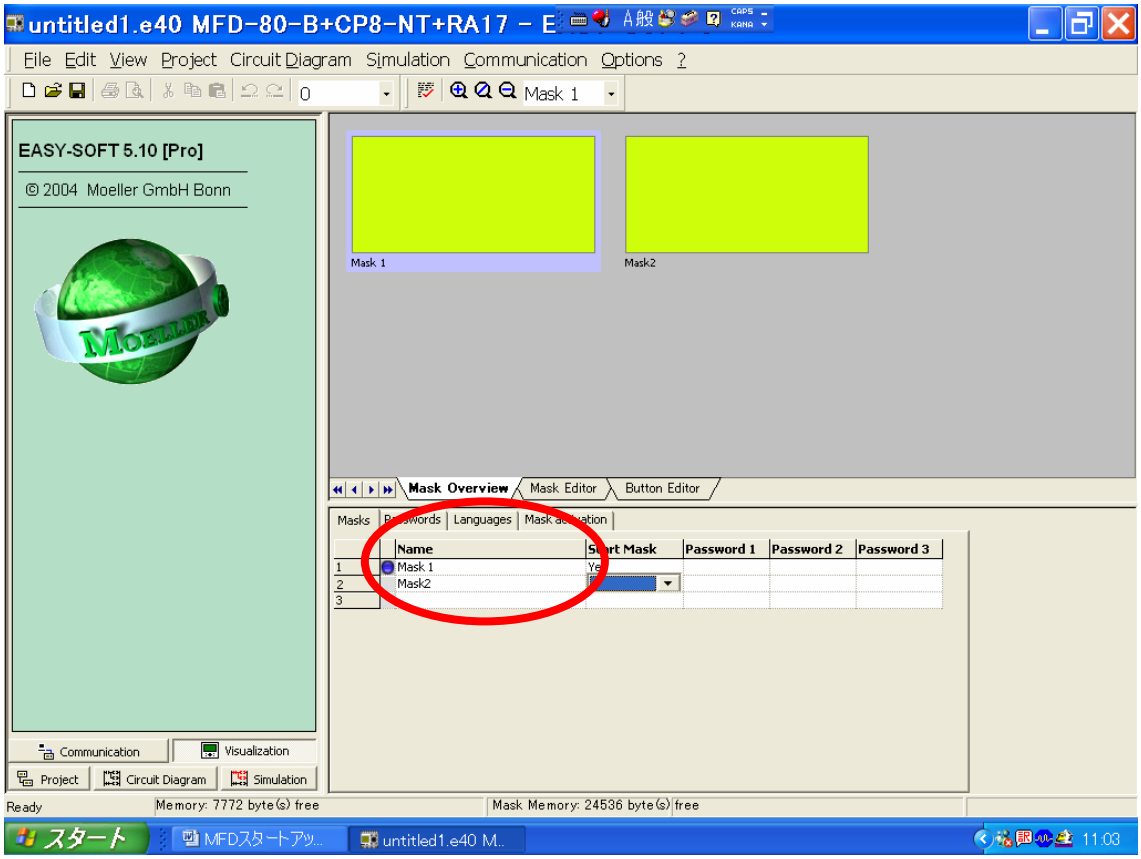


ここでは 9 枚のマスクを作成し、キーパッドの操作によって切り替わってゆくメニュー画面を作ってみましょう。

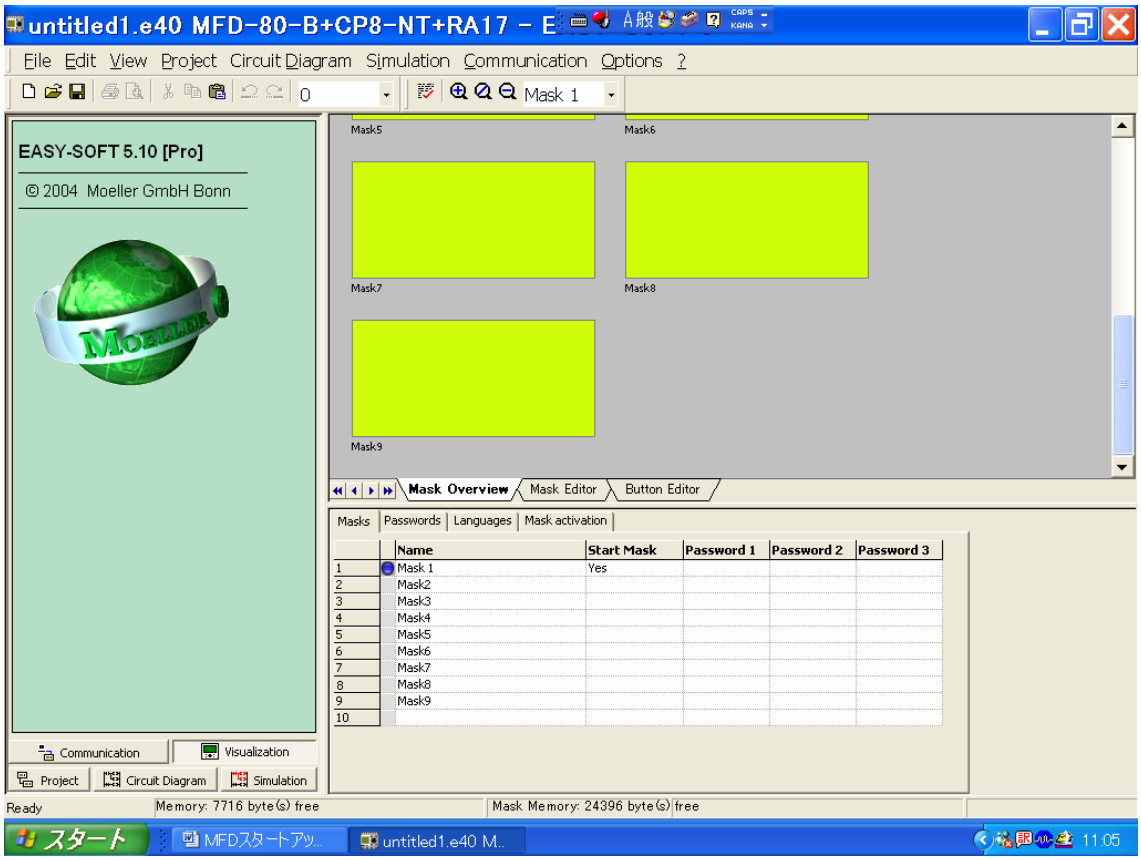
[タスク定義]現場のオペレータが 3 種類ある出力のうち、一つを選んで出力させるというメニューです。



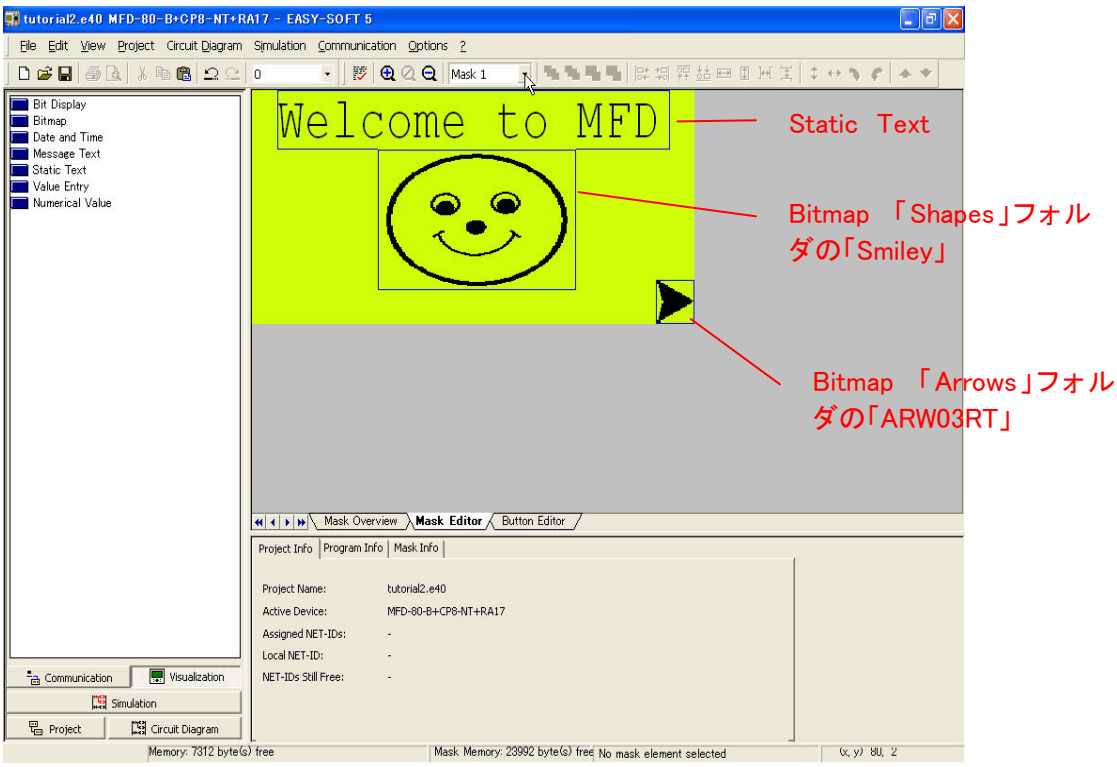
プロジェクト定義の後、「Visualization」ボタンを押します。マスクが 1 つ表示されます。



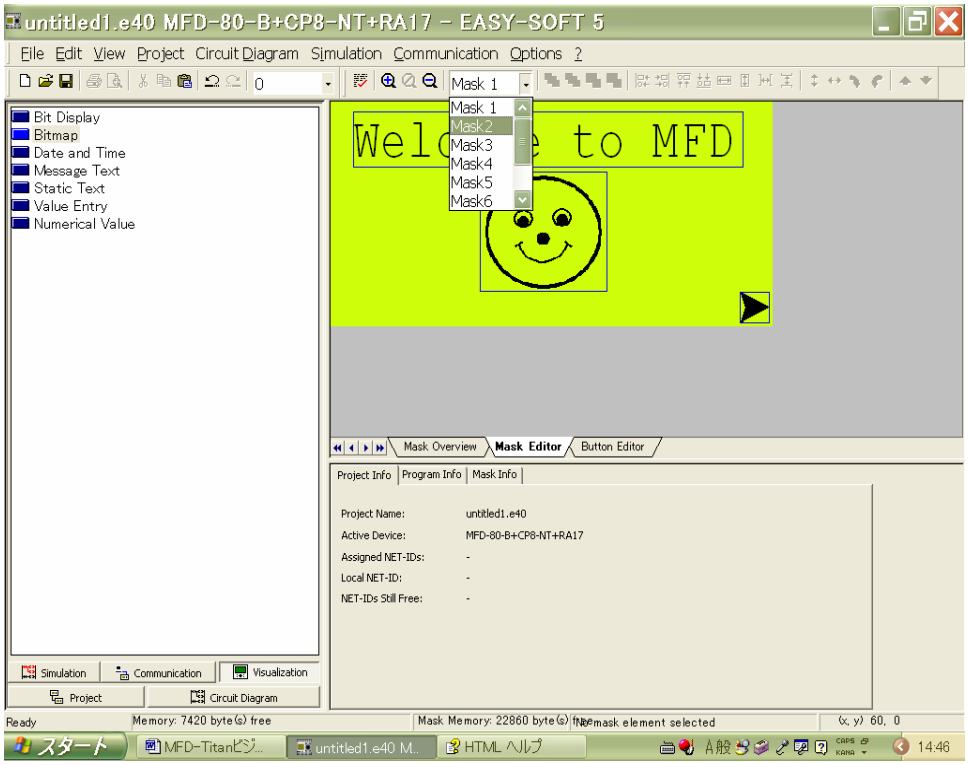
プロパティフィールドの Masks の Name に Mask2 と入力します。するとワークベンチにもう一つマスクができます。Start　Mask の欄はスキップして結構です。



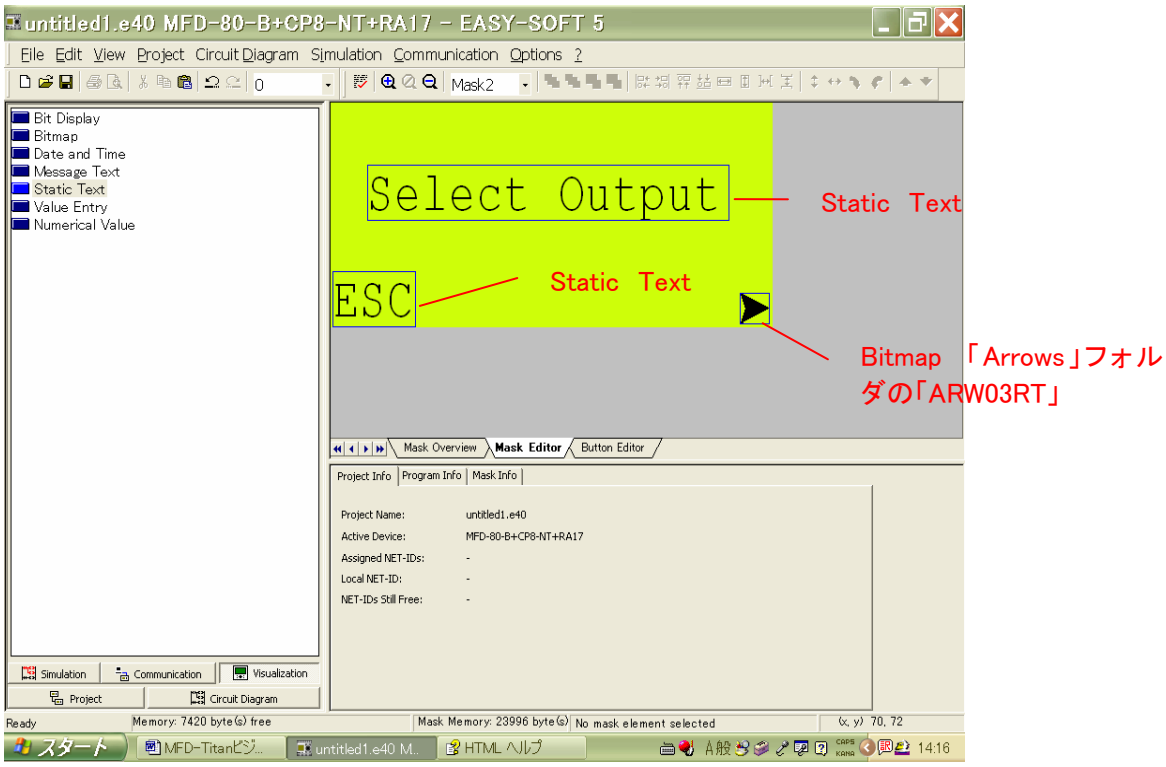
同様にして Mask9 まで作ります。Mask1 以外は Start　Mask 欄は空白です。



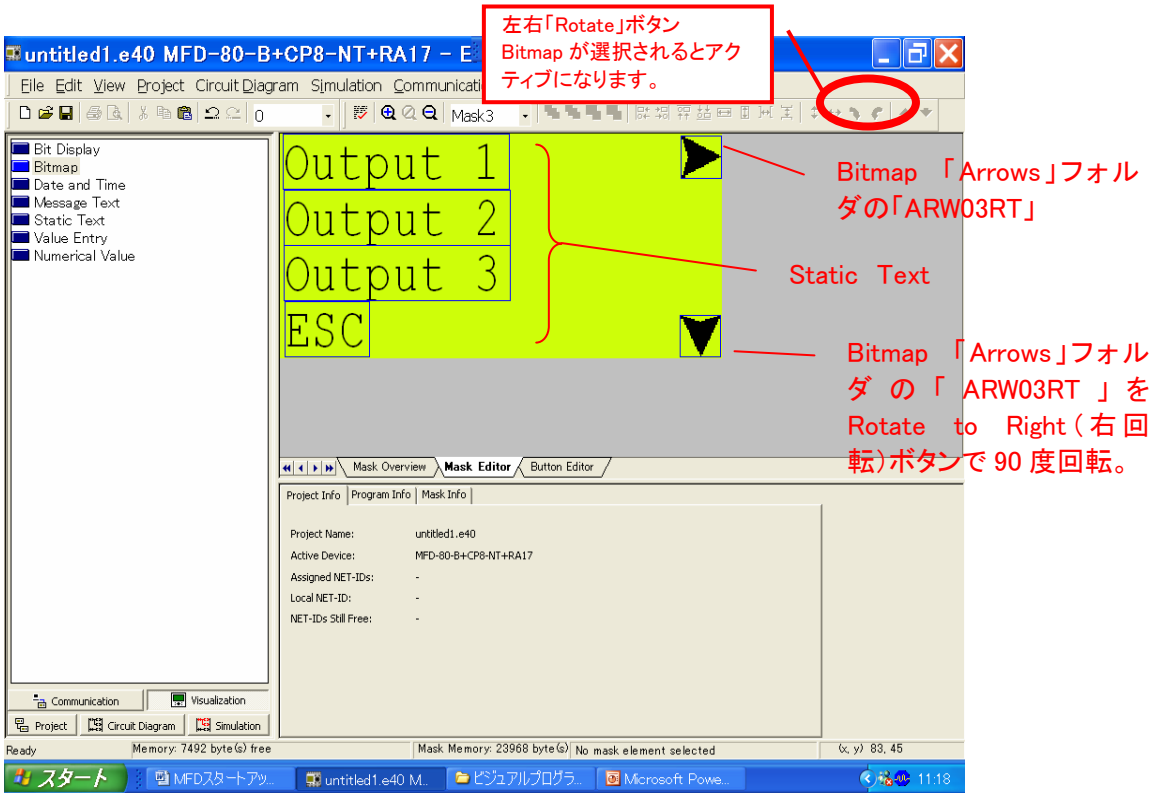
Mask1 をダブルクリックして編集画面にします。上図のように Mask1 を作ってください。



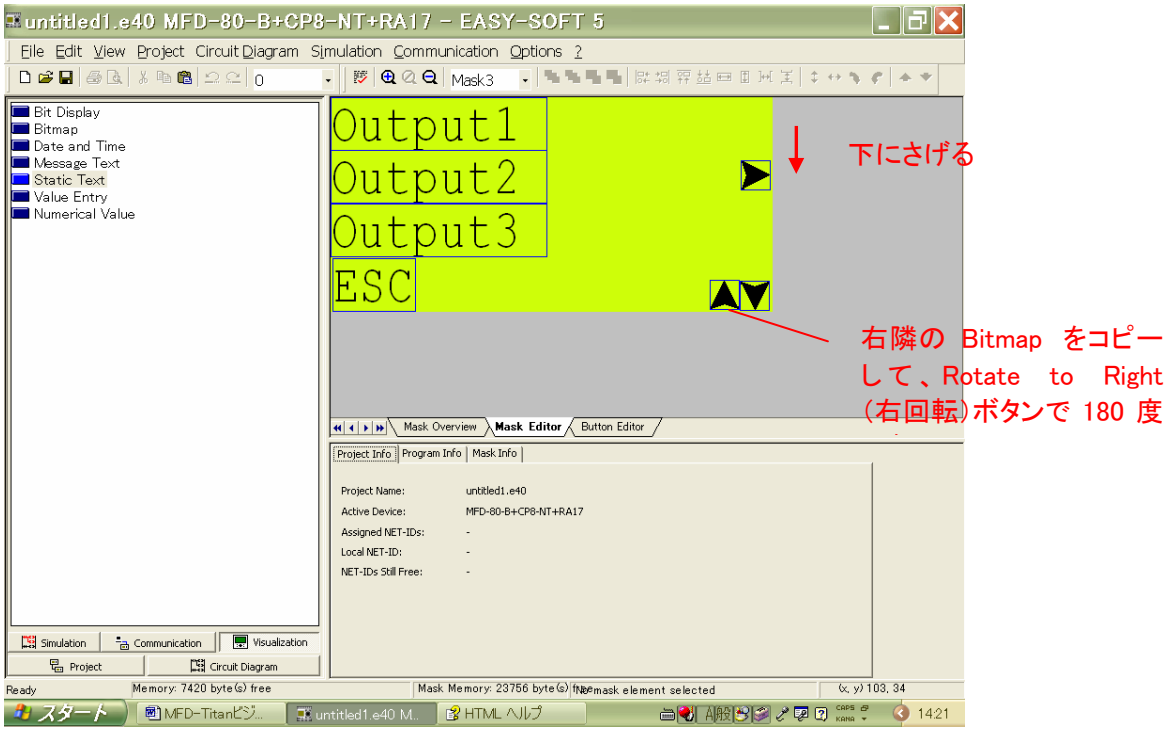
Mask のプルダウンメニューで Mask2 へ移ります。



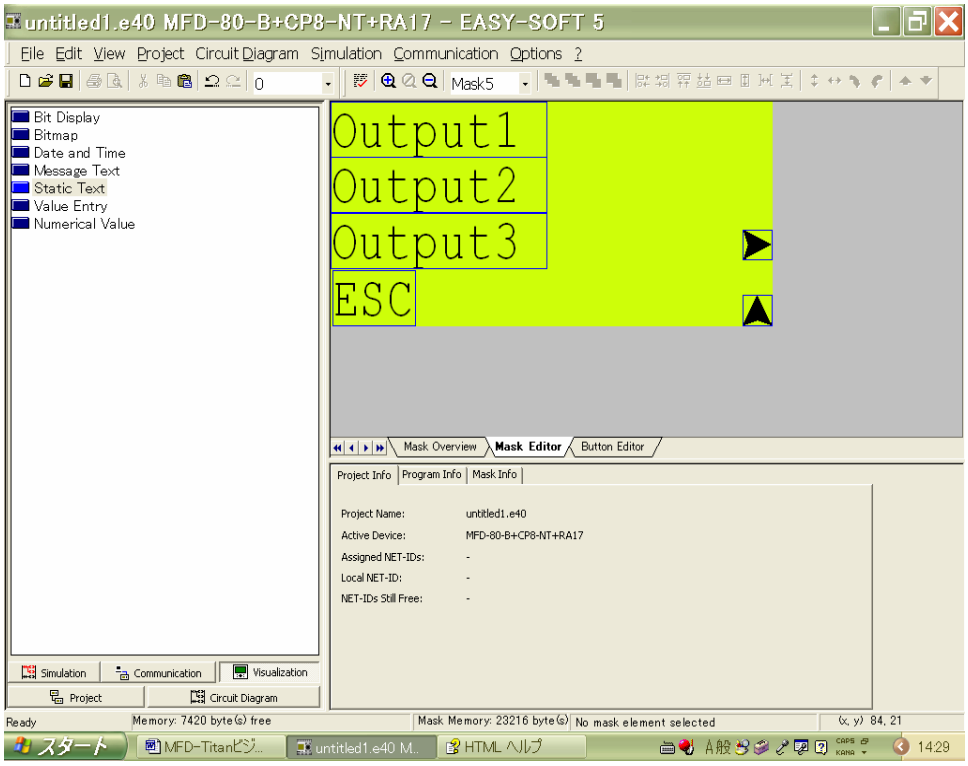
同様に上図のように Mask2 を作ります。



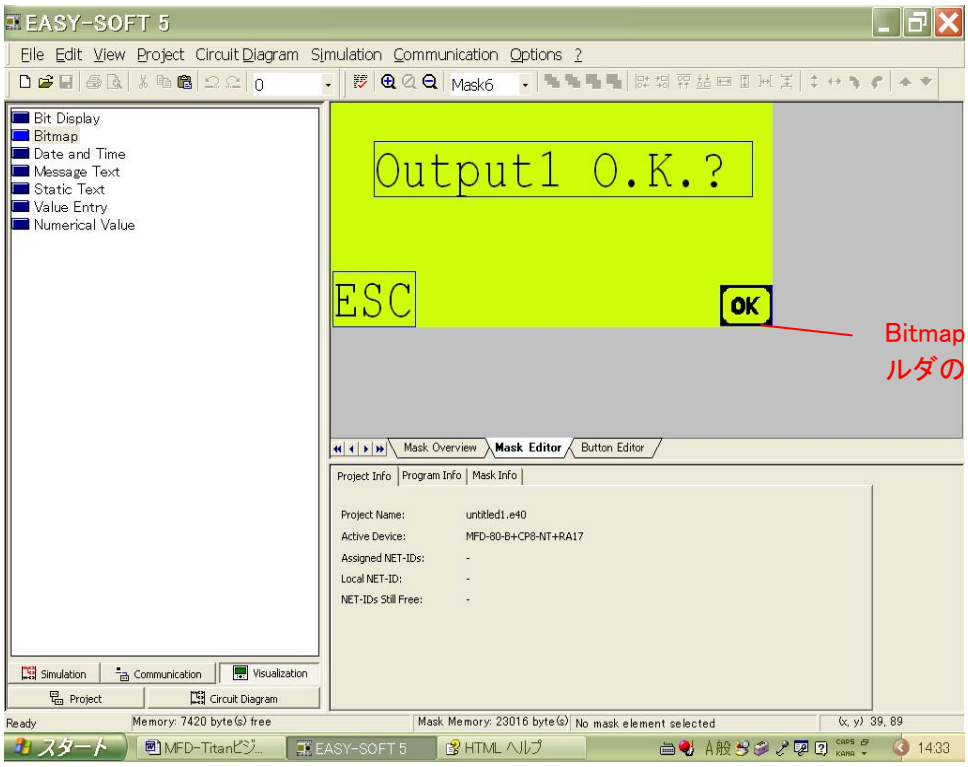
同様に上図のように Mask3 を作ります。矢印は右クリックか Edit メニューからコピーをし、90 度回転させるとすばやくできます。



同様に Mask4 を上図のように作りますが、Mask3 の画面で Edit→Select All、さらに Edit メニューで Copy をしてから Mask4 に移り、Edit の Paste で貼り付けます。その後位置を移動、不足分をコピー、回転で作るとすばやくできます。(Edit の代わりに全て右クリックでもできます。)

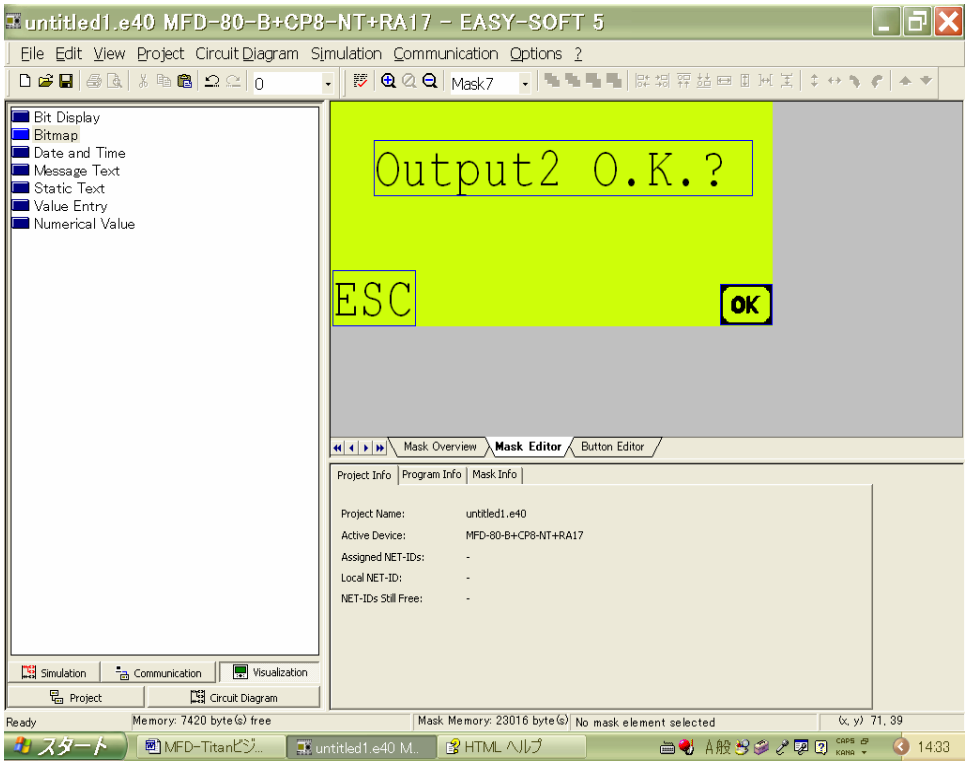


同様に Mask4 を全て選択(Select All)してコピー、Mask5 に貼り付けましょう。その後不要な部分を消去して上図のように Mask5 を作成します。

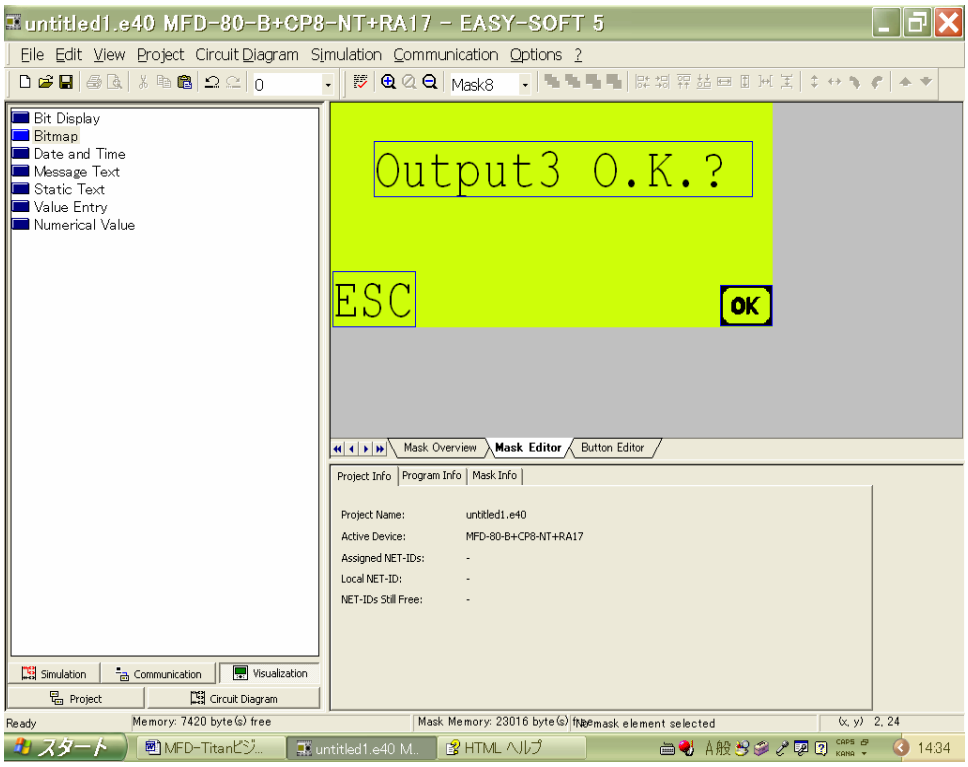


Bitmap「Icon Bitmap」フォルダの「OK」

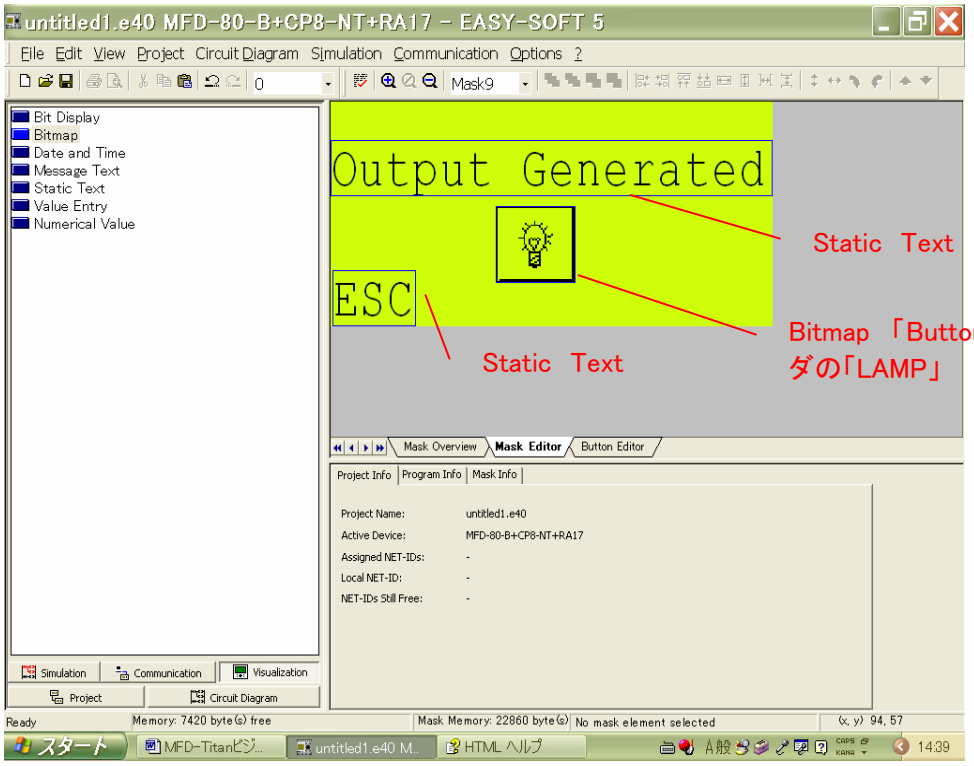
Mask6 は上図のように作成します。



Mask7 は Mask6 を全てコピーして貼り付けし、Output 1 の 1 を 2 に変えます。

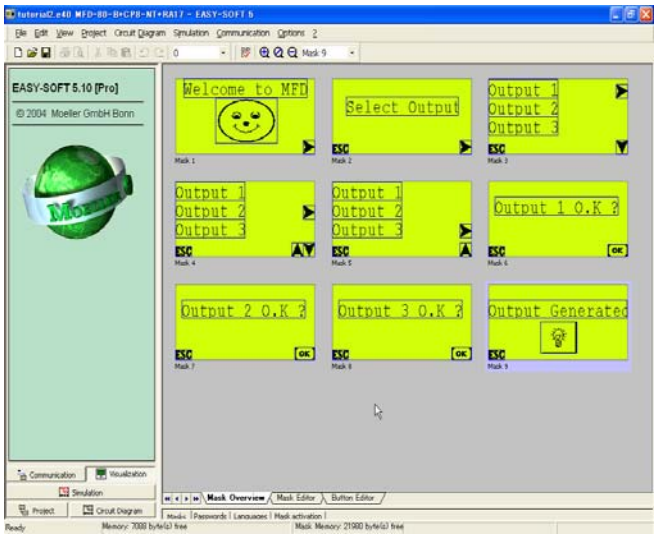


Mask8 も同様に Mask7 を全てコピーして貼り付けし、Output 2 の 2 を 3 に変えます。



最後の Mask9 です。以上のように作ります。これで、メニュー全画面の作成が終わりました。これらメニューにしたがって現場のオペレータが MFD-Titan のボタンで操作をします。次はその時各ボタンが担う役割を割り当てましょう。

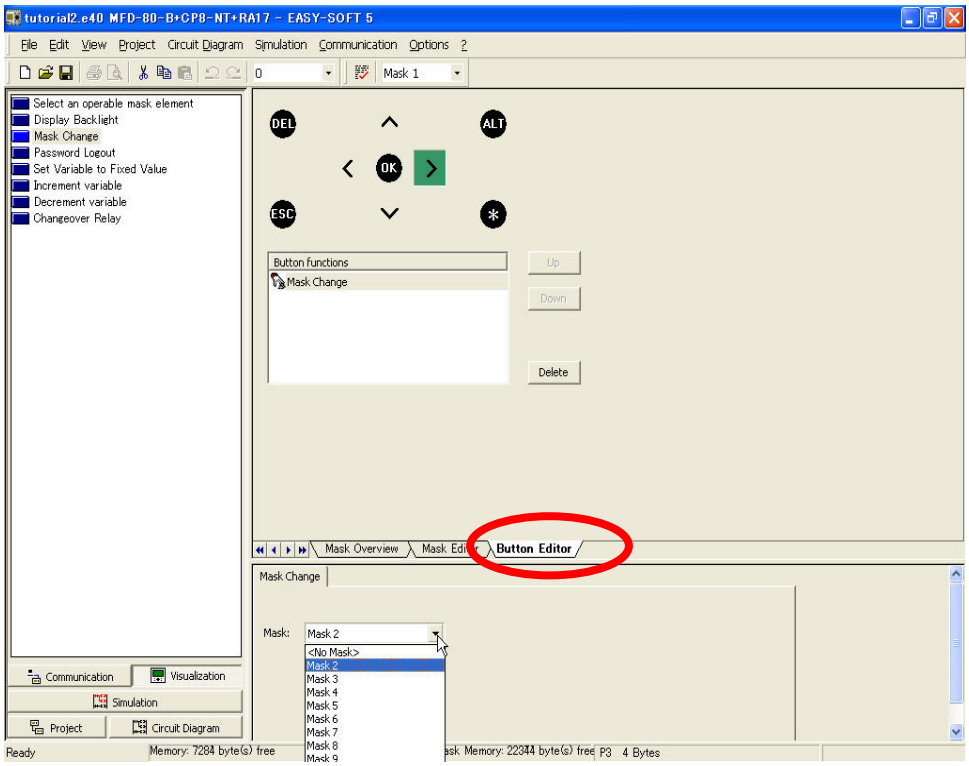
各画面によるボタンの役割



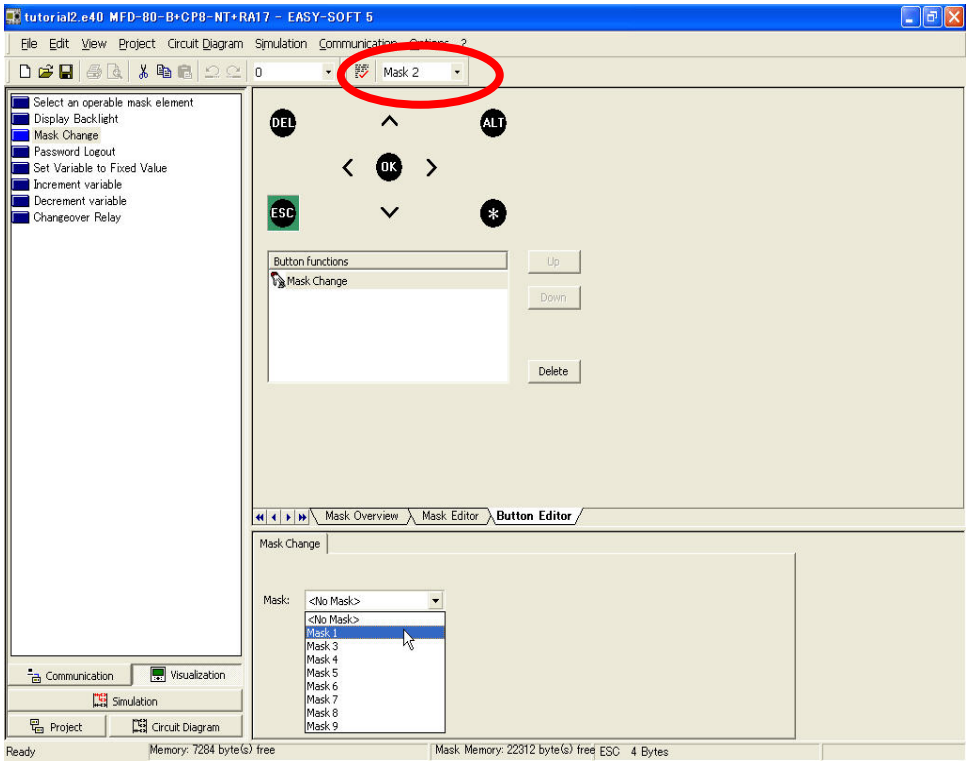
マスクによってボタンは以下のような働きをします。

Mask	▶	▼	▲	OK	ESC
1	メニュー画面に進む				
2	メニュー画面に進む				Mask1 に戻る
3	出力 1 を選択、Mask6 へ進む	他の出力を選択する画面へ進む			Mask1 に戻る
4	出力 2 を選択、Mask7 へ進む	他の出力を選択する画面へ進む	1 つ前の画面に戻る		Mask1 に戻る
5	出力 3 を選択、Mask8 へ進む		**1つ前の画面に戻る**		Mask1 に戻る
6				選択完了、出力 1 に出力、Mask9 へ	Mask1 に戻る
7				選択完了、出力 2 に出力、Mask9 へ	Mask1 に戻る
8				選択完了、出力 3 に出力、Mask9 へ	Mask1 に戻る
9					Mask1 に戻る

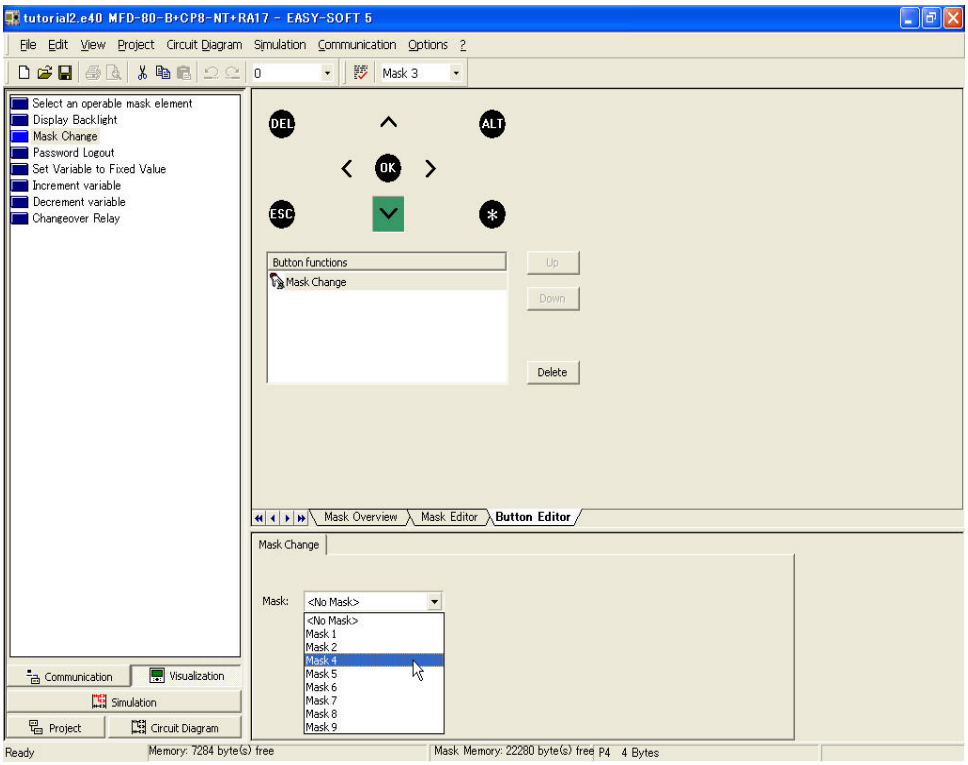
以上のような機能を MFD-Titan のボタンに割り当てていきましょう。



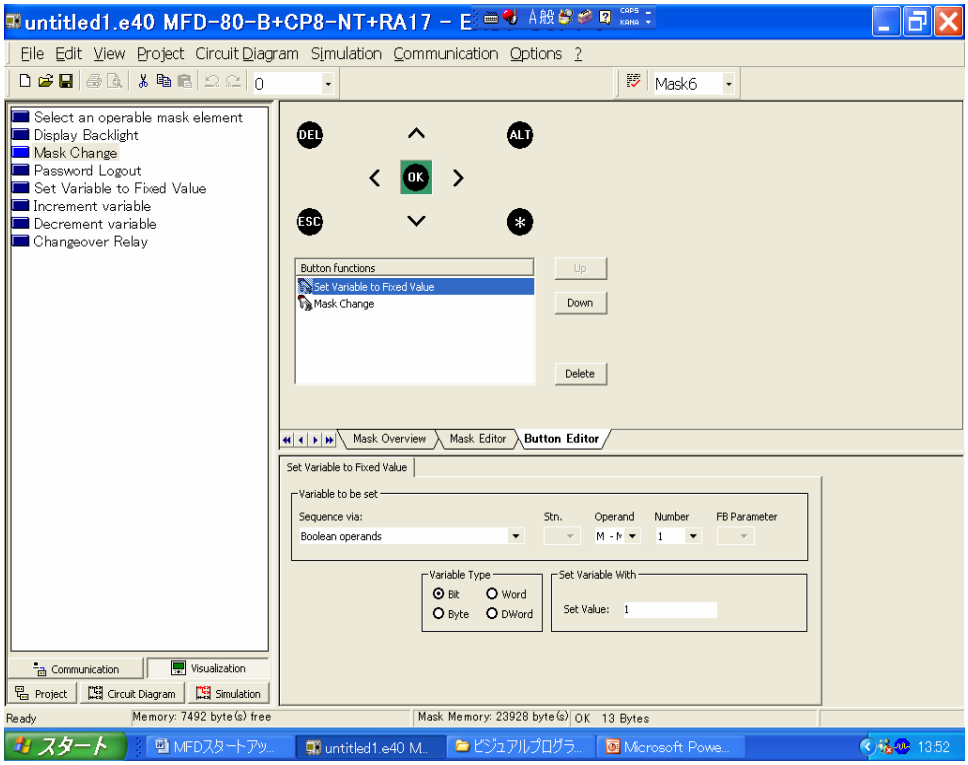
「Bottun Editer」を押します。ツールボックスにはボタンに割り当てられる機能が表示され、ワークベンチにはボタンのダミーが表示されます。Mask1 の右向き矢印を選択し、ツールから「Mask Change（マスク切替）」をワークベンチの Button Function 下の空白にドラッグします。プロパティフィールドの Mask Cahnge のタブでは、移動先のマスクを設定します。ここでは Mask2 です。



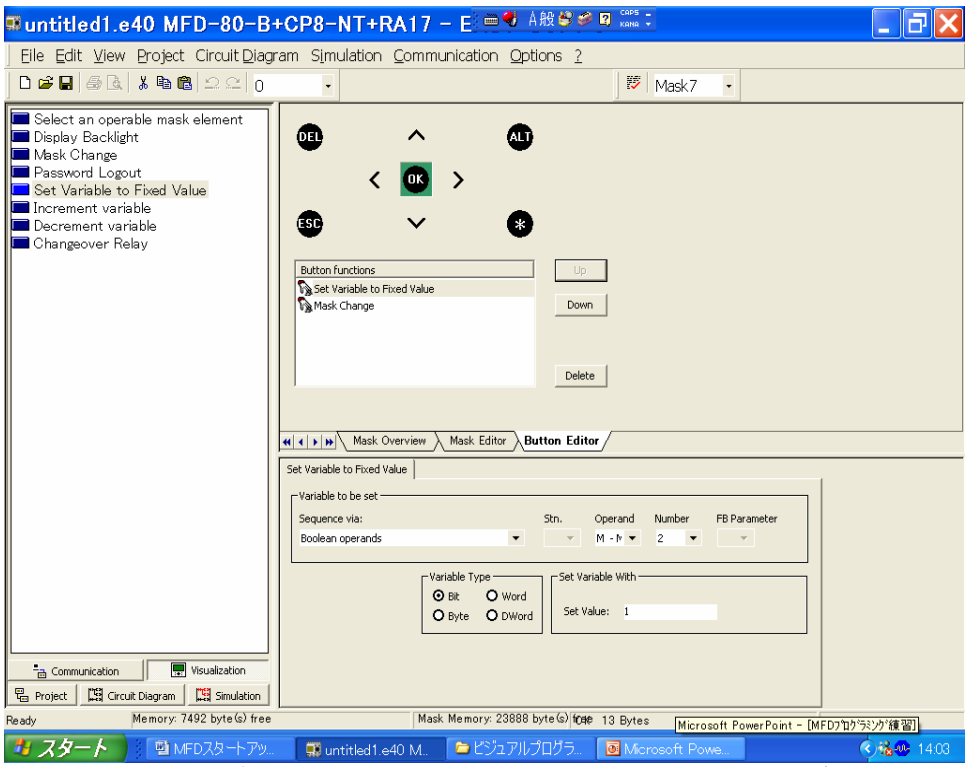
Mask2 での ESC ボタンでは一つ前の Mask1 に戻る「Mask Change」を割り当てます。右向き矢印にも Mask1 同様の設定をしてください。



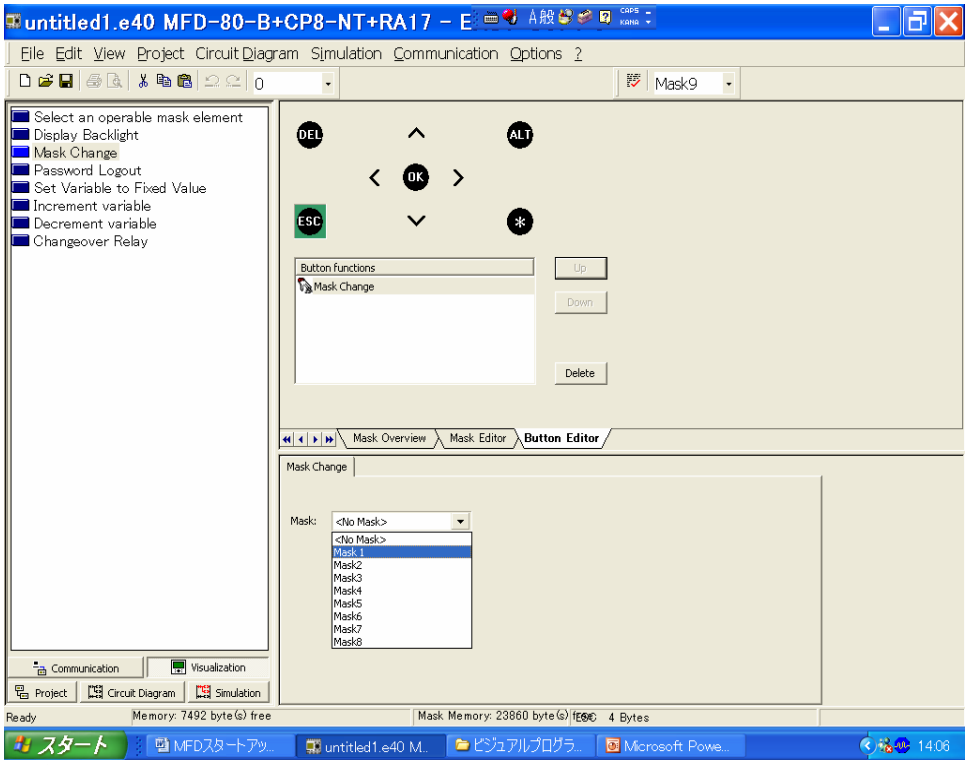
同様に Mask3 にも、49 ページの各ボタンの機能に従って 3 つのボタンに機能を割り当ててください。



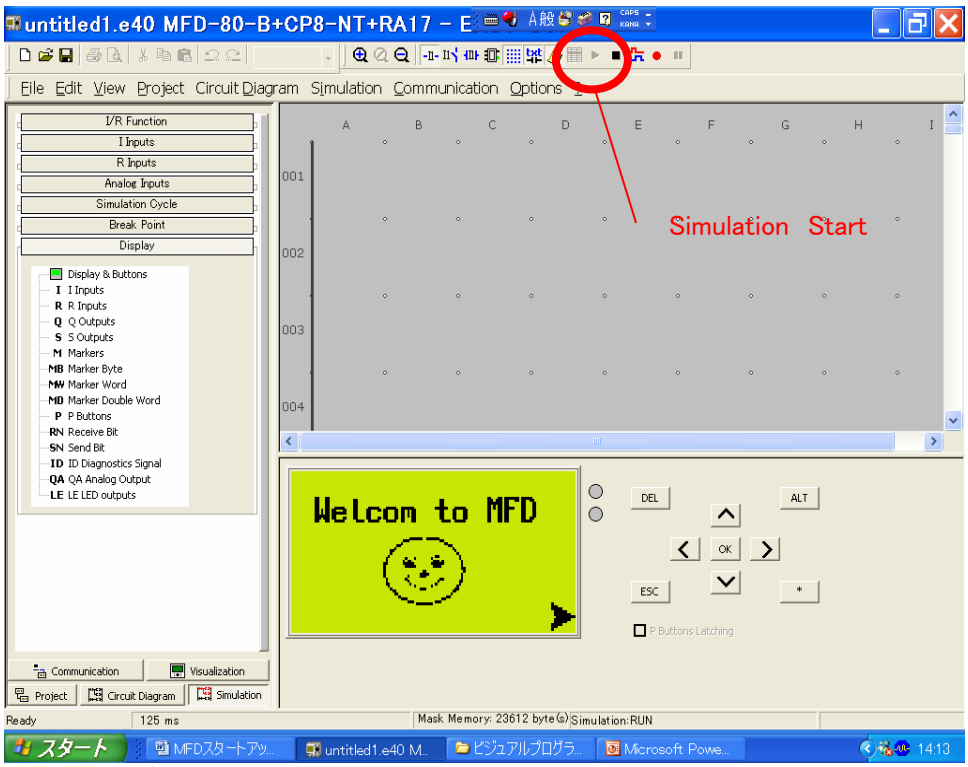
Mask6 の OK ボタンには「Mask Change」と共に、「Set Value to Fixed Value」を割り当てます。Variable Type を Bit に選ぶ、Sequence via は Boolean Operands にします。ここでは Operand を M-Marker、Number を 1 に選びます。また Set Value を出力させる「1」に設定します。



同様に Mask7 にもボタンの役割を割り当てます。マーカナンバーはラダー回路に合わせて、適宜つけてください。



最後の Mask9 の ESC の設定をしたら終了です。
シミュレーションをしてみましょう。



Simulation ボタン→Display→Display&Button→Start Simulation を押します。
プロパティフィールドのボタンをクリックして動作を確かめてみましょう。

正確に画面が切り替わりましたか？

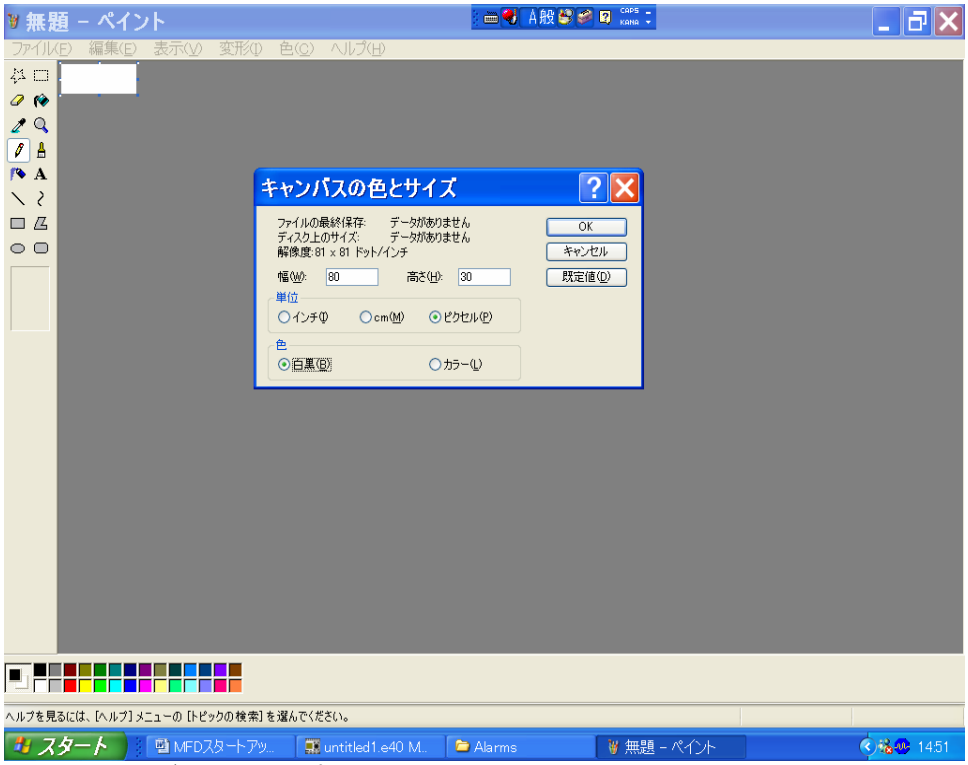
以上のビジュアル画面は、実際にユーザが必要とするラダー回路と連動して応用されます。
そのラダー回路に合わせて、マーカの番号や設定値を設定してください。

８． 日本語のビットマップ作成

残念ながら、現在のところ MFD-Titan のテキストでは日本語の入力はできませんが、ビットマップとして日本語のメッセージを比較的簡単に作って使用することも可能です。その際、「ペイント」などのお絵かきソフトでビットマップファイル（.bmp）を作成できるアプリケーションが必要になってきます。「ペイント」は通常、Windows のアクセサリに添付されています。

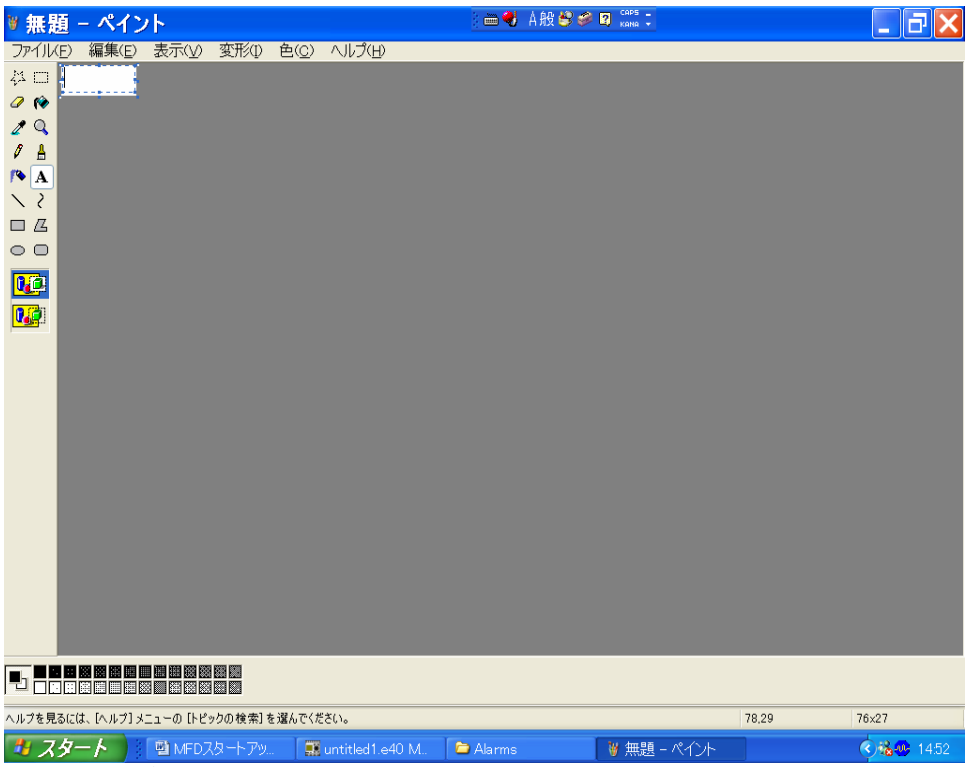
ここではその方法を、「ペイント」を使用してご紹介します。

ペイントを立ち上げます。
変形メニューのキャンバスの色とサイズをクリック。



ここではサイズを 80×30 ピクセルにして、白黒にします。
白黒にするとカラー情報が失われますが実行しますか？と聞いてきますが、Yes で確定します。

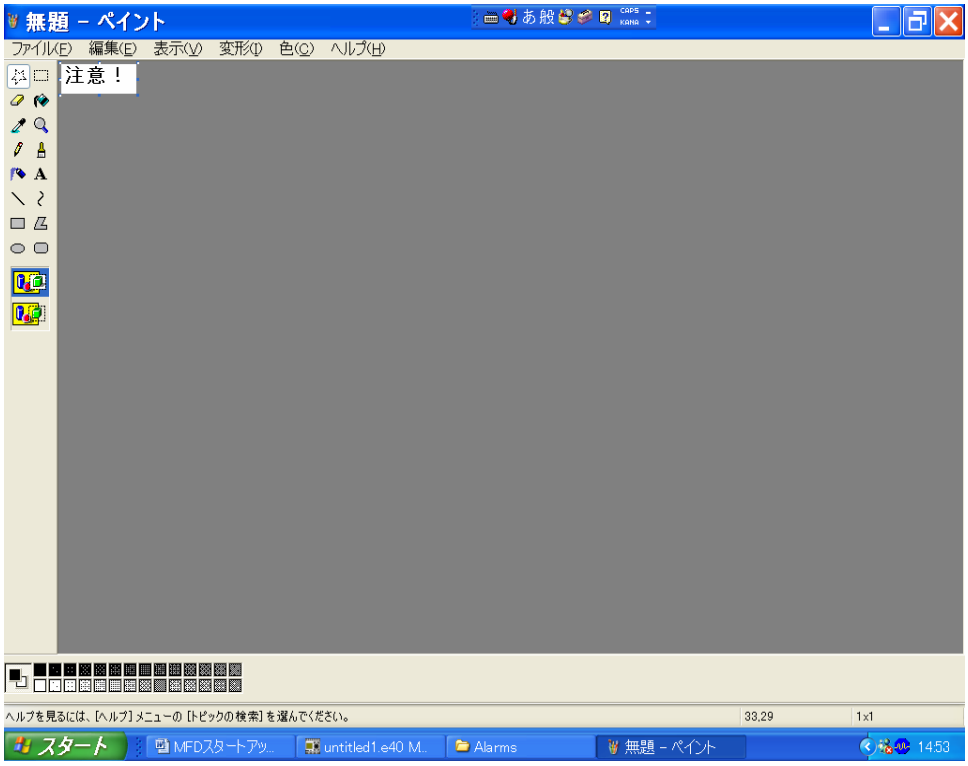
参考　MFD-Titan のマスクは、全部で 24Kbyte の容量を持っています。7 節のようにたくさんの画面を切り替えていくようなプログラムの場合、一つのビットマップの大きさを 400 から 500 バイトに抑えておくことをお勧めします。この説で作る漢字のビットマップは、上記のサイズで 422 バイトになります。必ず白黒設定をしてください。



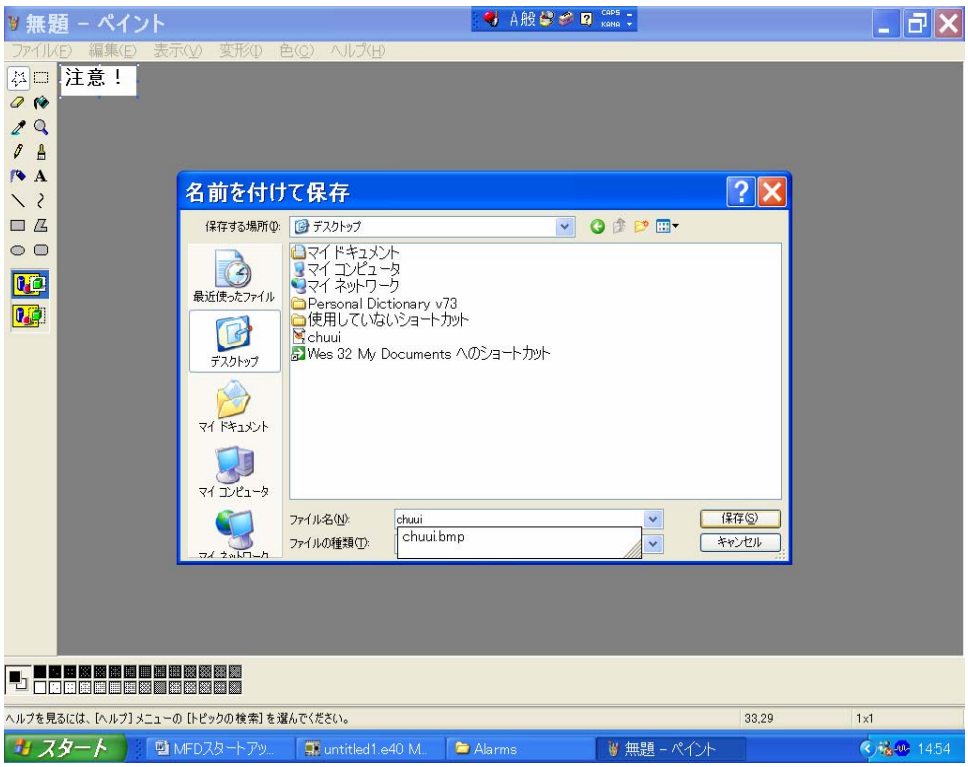
キャンバスができました。

「A」　　の文字ボタン（テキスト入力ボタン）をおして、キャンバスいっぱいに枠を作ります（十字のポインタになりますので、それをドラッグしてください）。

注意　表示メニューで拡大すると「A」テキスト入力機能は使えませんので、標準の表示のまま、作業を行ってください。

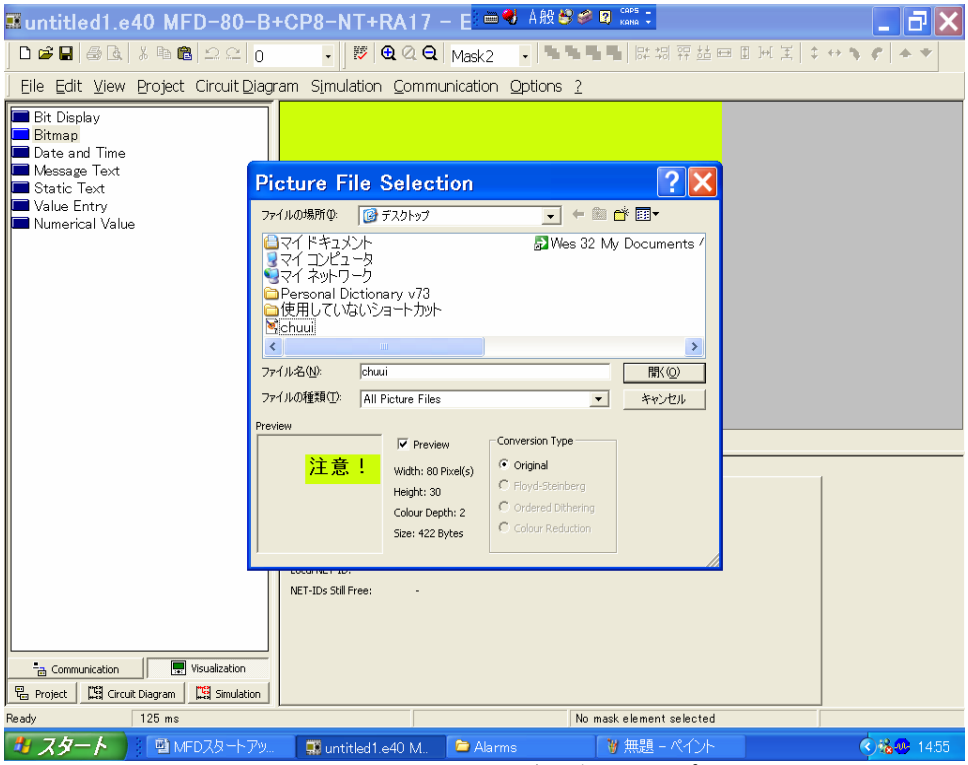


漢字で「注意！」と入力します。

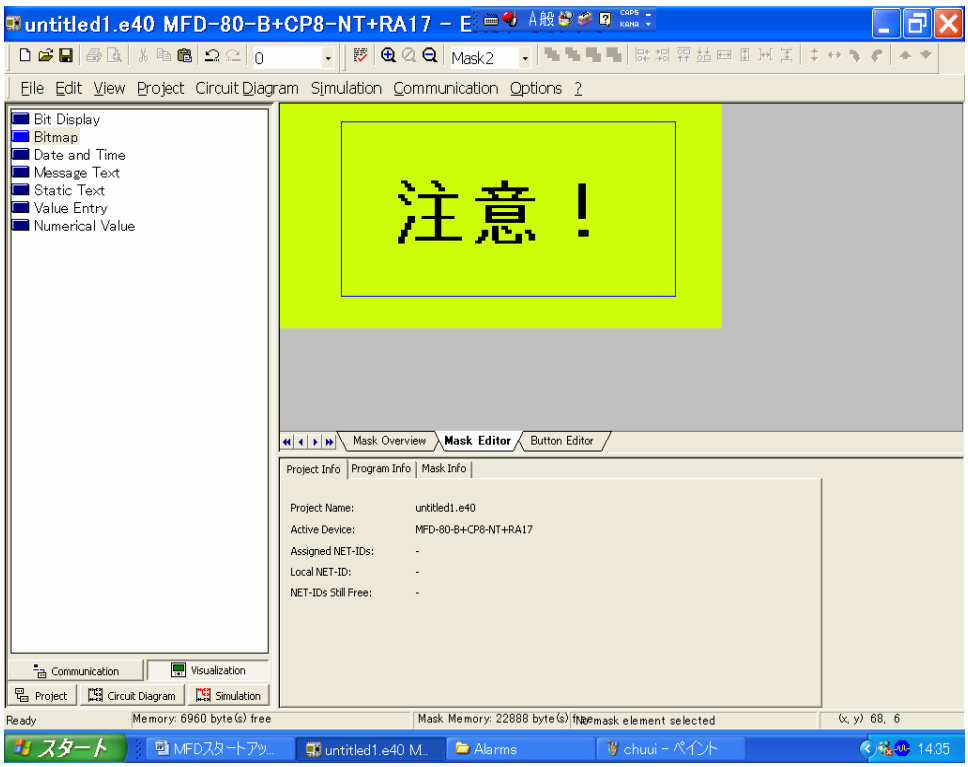


ファイルメニューの「名前を付けて保存」でわかりやすい場所に保存します。
ここではデスクトップに「chuui」の名で保存します。

EASY-SOFT V.5.10 Pro　の Visualization 画面に戻ります。



適当なマスクを作り、Bitmap　ツールをドラッグ。デスクトップから先ほど作った漢字ビットマップを選択します。



以上のように日本語の表示ができます。

お客様のニーズにあった使い方をしていただけたら、独自の最適なソリューションを創造していただけるものと確信しております。

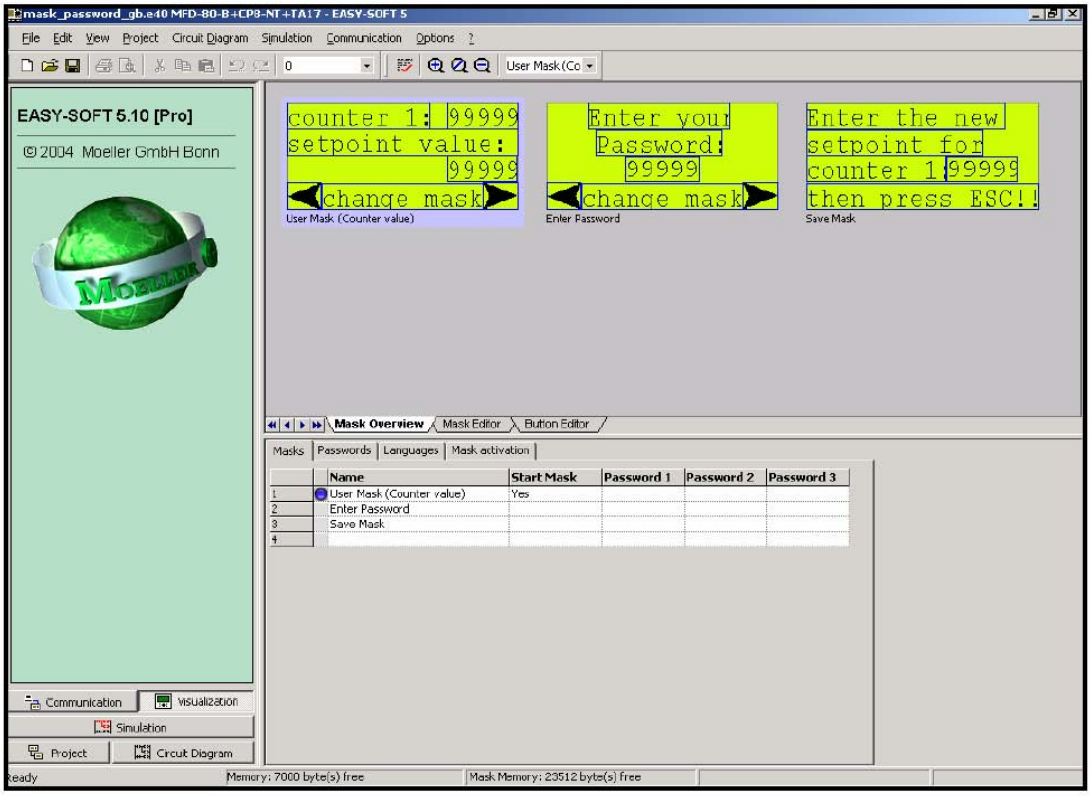
以上は MFD-Titan のビジュアルプログラムについてのご紹介でした。
なにかご不明の点がありましたら、お近くのセールスエンジニアまで、
お気軽にお尋ねください。

９． 例：パスワード付き入力画面（プログラムがダウンロード可能）

実践プログラム：“mask_password_GB.e40” による実用例

ここでは弊社ホームページからダウンロードが可能な、EASY-SOFT V5.10 PRO 用のプログラム“mask_password_GB.e40”について解説します。以下のようなタスク定義で、お客様が実際にそのままご利用いただくことも可能です。

[タスク定義] 以下のように 3 つのマスクから成ります。マスク1はカウンタの実行値と設定値を表示、マスク 2 でパスワードを入力。パスワードが正しければ、自動的にマスク 3 に移行し、カウンタの設定値を変更入力できる。



“mask_password_GB.e40”は弊社ホームページ
http://www.jpn-moeller.co.jp/
MFD-Titan 多機能ディスプレイのページからダウンロードできます。

このプログラムは、本書冒頭でも説明してあるように、
① ビジュアライゼーション（Visualization）部分
② ラダー回路
の 2 つの部分から成っています。どちらが欠けてもプログラムは作動しません。
では、プロジェクト定義は済んでいますので、ビジュアライゼーション（Visualization）とラダー回路作成について順を追って見ていきましょう。

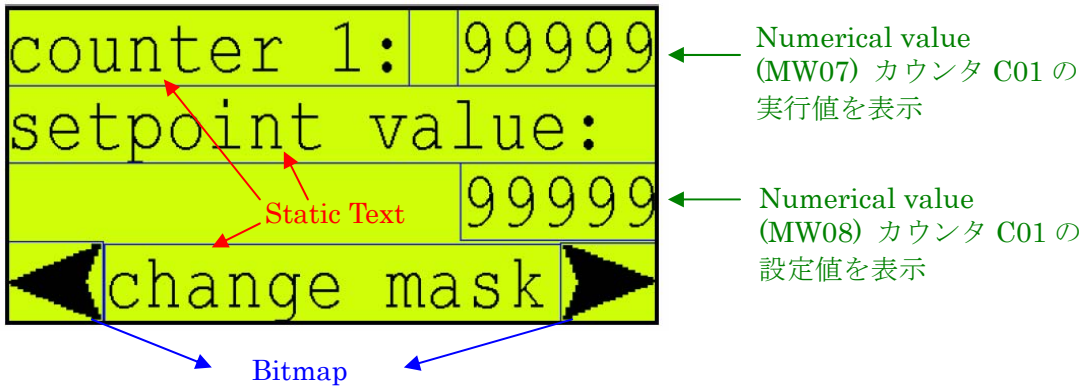
ダウンロードしたプログラムはすでに完成していますが、作成プロセスを見るように解説していきます。

①マスクのビジュアライゼーション（Visualization）

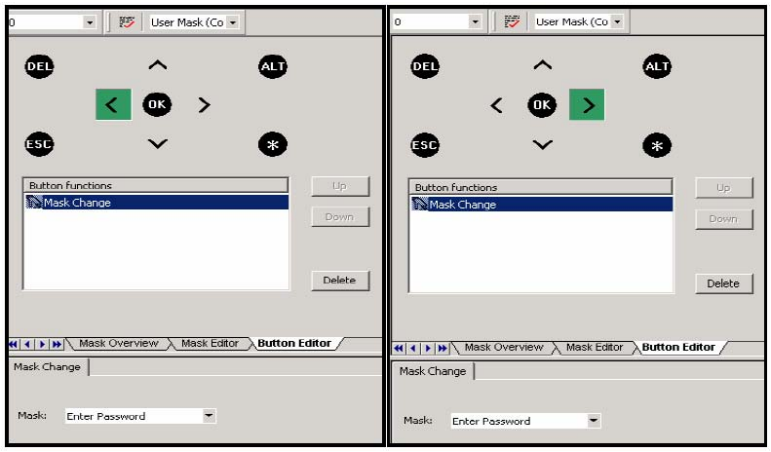
それぞれのマスクの構成や編集は以下のようになっています。

マスク1：
マスク1は”User Mask（Counter value）”と名づけられ、カウンタ C01 の現在の実行値と設定値を表示しています。また、マスク 2 に切り替える方法も表示しています。ここでは P1 ボタン（左向きカーソルボタン）か P3 ボタン（右向きカーソルボタン）でマスク 2 に移動します。

マスクエディタ（Mask Editor）での編集



ボタンエディタ（Button Editor）での編集

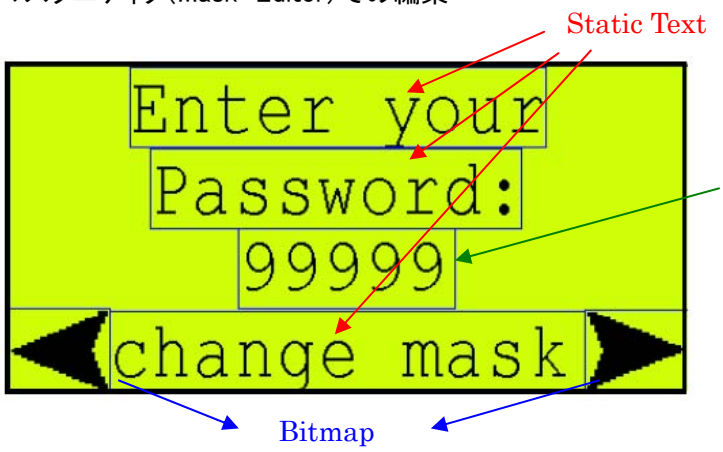


P1 ボタン（左向きカーソルボタン）とP3ボタン（右向きカーソルボタン）に「Mask Change」の機能を割り当て、移動先を「Enter Password」（マスク 2）にします。

マスク 2:

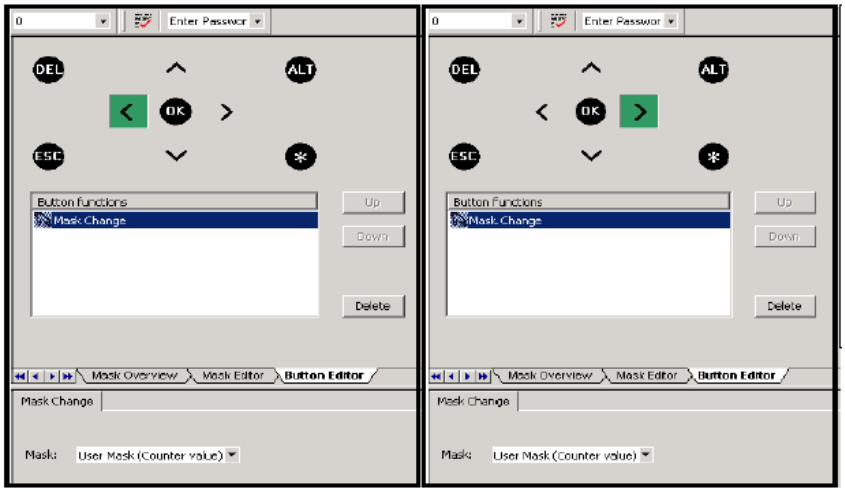
マスク 2 は「Enter Password」という名前です。ここではユーザにパスワードを入力するよう要求する画面で、パスワードが正しければマスク 3(「Save Mask」)に自動的に移行する機能を持っています。また、マスク 1 に切り替える方法も表示しています。ここでは P1 ボタン(左向きカーソルボタン)か P3 ボタン(右向きカーソルボタン)でマスク 1 に移動します。

マスクエディタ(Mask Editor)での編集

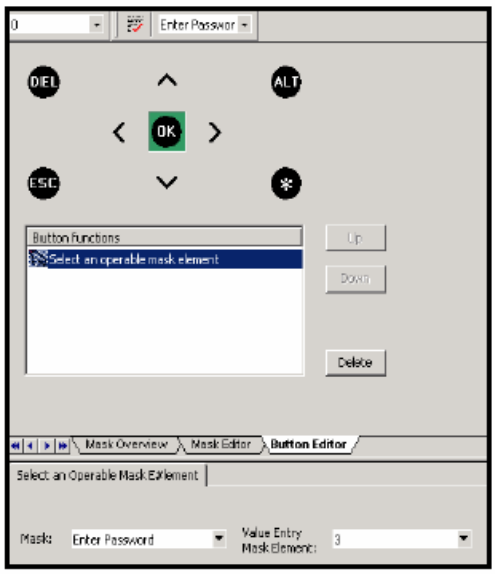


Value Entry (MW09) ユーザはパスワードを入力。
これは OK ボタンで入力可能な状態になります。ユーザはパスワード入力後、再び OK ボタンを押します。

ボタンエディタ(Button Editor)での編集



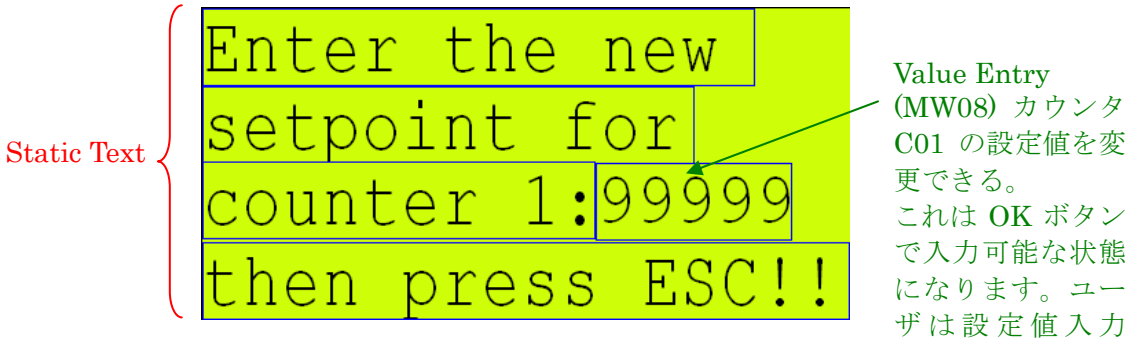
P1 ボタン(左向きカーソルボタン)と P3 ボタン(右向きカーソルボタン)に「Mask Change」の機能を割り当て、移動先を「User Mask(Couter value)」(マスク 1)にします。



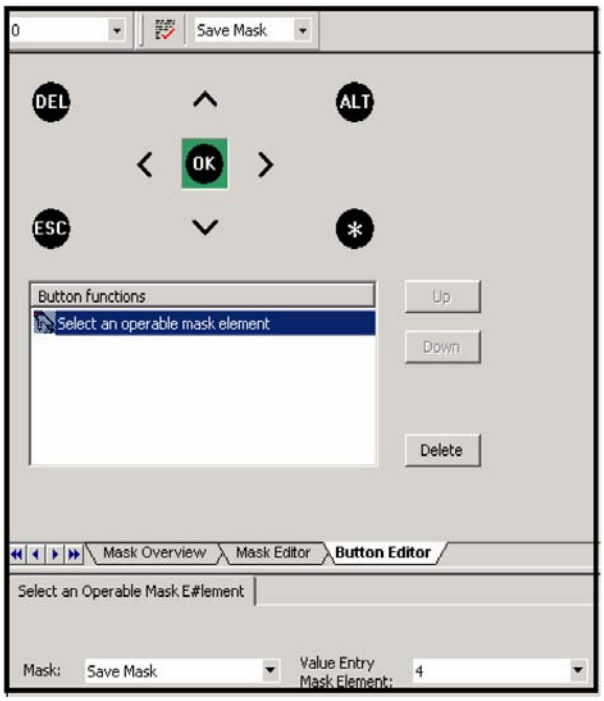
忘れてはいけないのが、OK ボタンへの機能の割り当てです。OK ボタンには「Select operable mask element」という機能を割り当てます。これはマスク上に入力された要素をプログラムに取り込むという機能です。
入力ができるマスクの要素は1つだけ、すなわち 3 番目の「Value Entry(99999 の表示)」だけですので、Value Entry Mask Element のドロップダウンメニューには「3」が自動的に入っています。これを選択します。

マスク 3:
マスク 3 はいわば”隠され、保護されたマスク”です。マスク 2 で入力されたパスワードが正しくない限り、表示はされません。正しいパスワードが入力され、マスク 3 が表示されるとユーザはカウンタの設定値を変更することができます。
設定変更が終了した後のために、ESC キーで画面を抜けられるようにします。

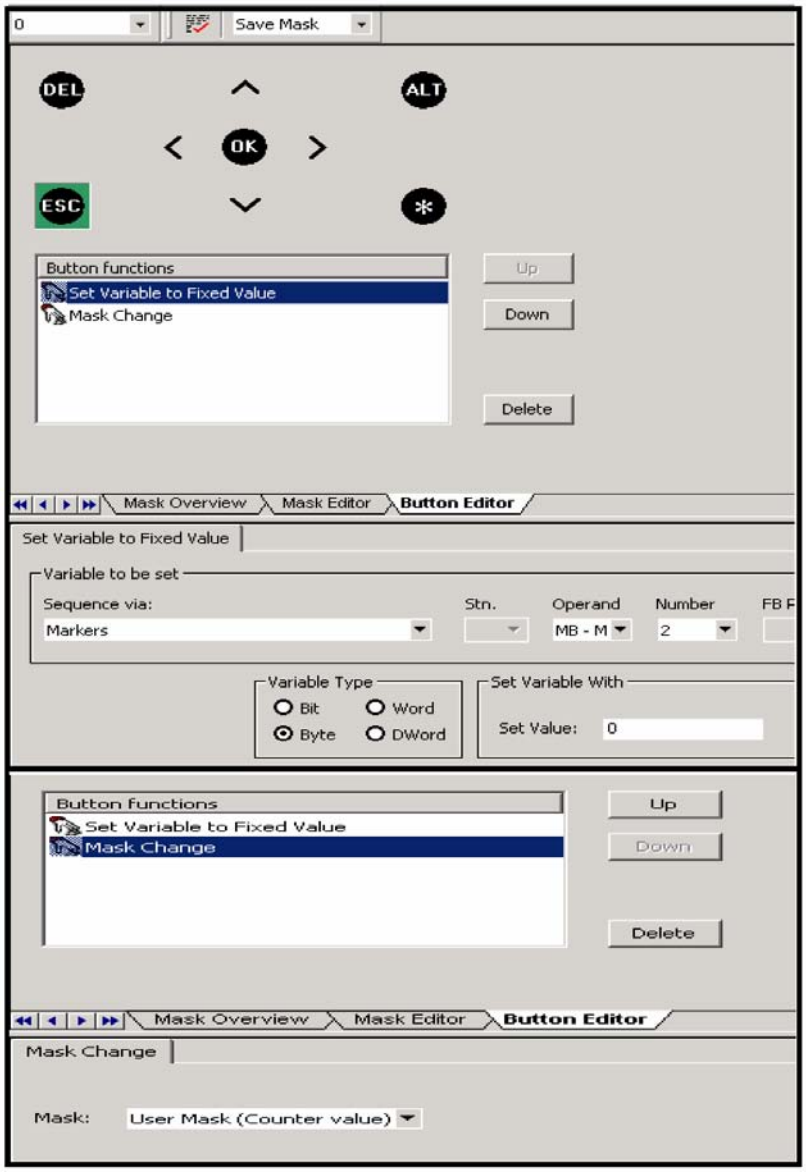
マスクエディタ(Mask Editor)での編集



ボタンエディタ(Button Editor)での編集



マスク 2 と同様に、OK ボタンへ「Select operable mask element」という機能を割り当てます。これはマスク上に入力された要素をプログラムに取り込むという機能です。
ここでの入力ができるマスクの要素は1つだけ、すなわち 4 番目の「Value Entry (99999 の表示)」だけですので、Value Entry Mask Element のドロップダウンメニューには「4」が自動的に入っています。これを選択します。



ESC ボタンには 2 つの機能が割り当てられます。

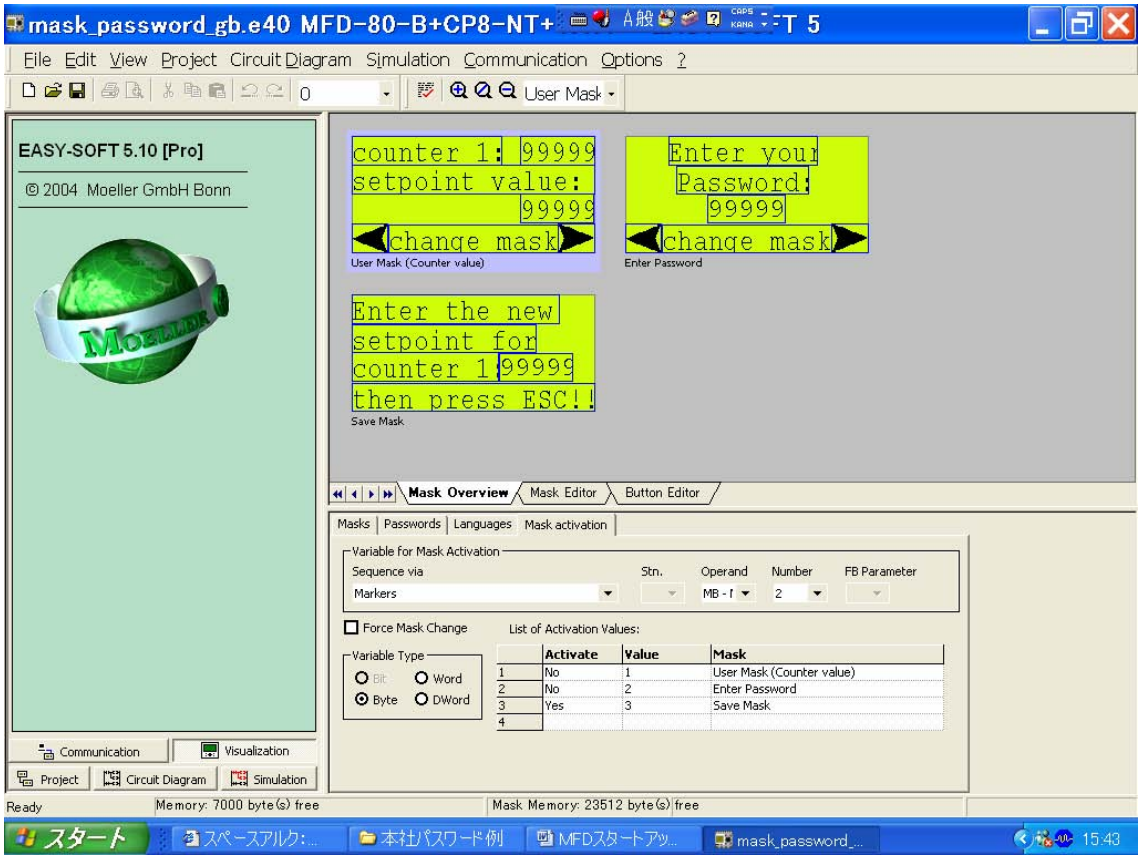
1)「Set Variable to Fixed Value」で、マーカバイト MB02 の値を 0 にします:

これには今まで表示されていたマスク 3 を画面から消す作用があります。詳しくは次の「マスクコントロール」の項と、ラダー回路および回路内のデータファンクションブロック DB01 の項を参照してください。

2)マスクの切替:

最初のマスク 1 に戻ります。

②マスクのコントロール



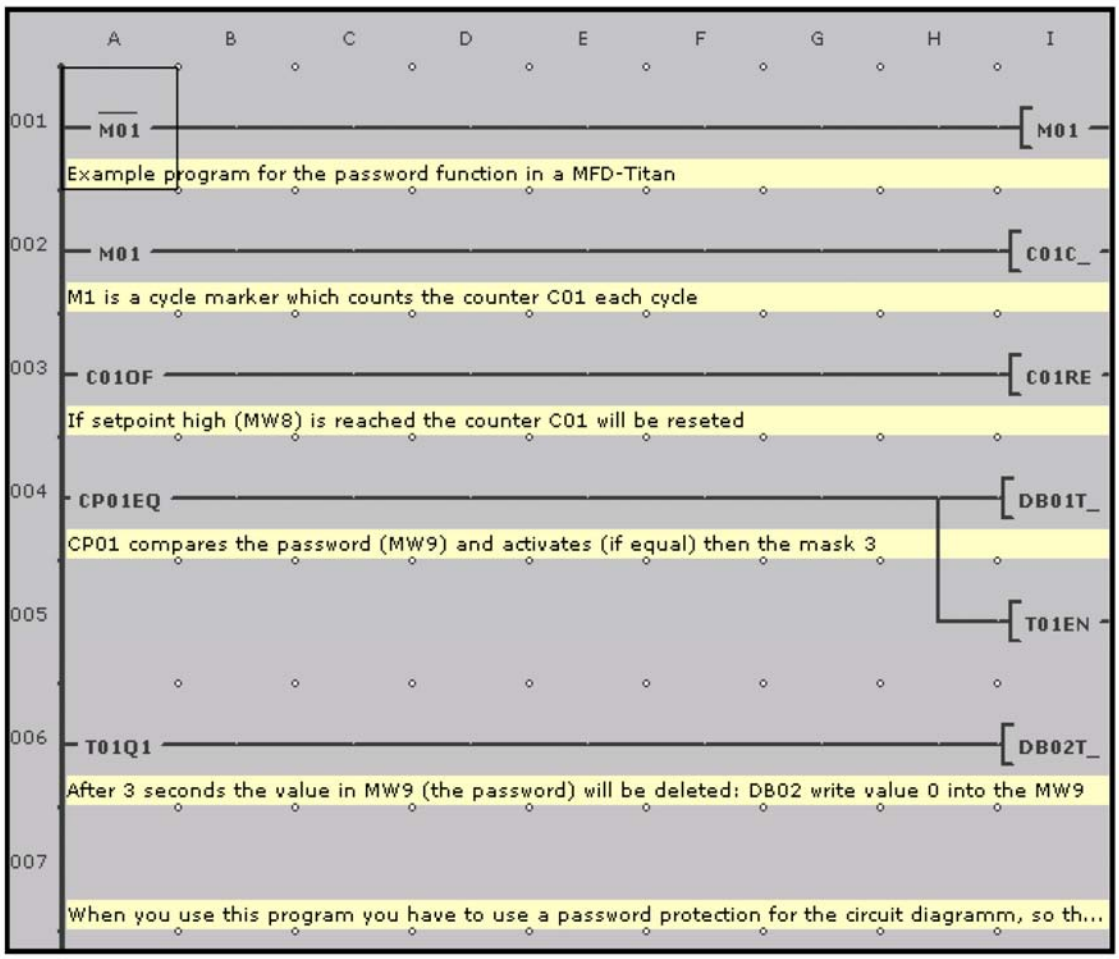
マスク 3 への切り替えはラダー回路内のマーカバイト MW02 によって行われます。パスワードが正しければ MW02 がマスク 3 に割り当てられた値 3 を出力するようにし、それによってマスク 3 が表示されるように設定します。
マスクには作成された順によって、値が 1 から始まる連番で自動的に割り当てられます。ワークベンチの「Mask Overview」タブを開き、プロパティフィールドで「Mask activation」を開くと確認できます。

ここで必要なのは、ワークベンチでマスク 3 を選択した状態で、プロパティフィールドの「Sequence via」で「Markers」を選択し、「Operand」を「MB-Marker Byte」に選択し、「Number」を「2」にします。これでマスクが表示されるには MW02 の値 3 出力が必要になるように設定できました。

プログラムの全体像を見るには、次項のラダー回路作成をご参照ください。

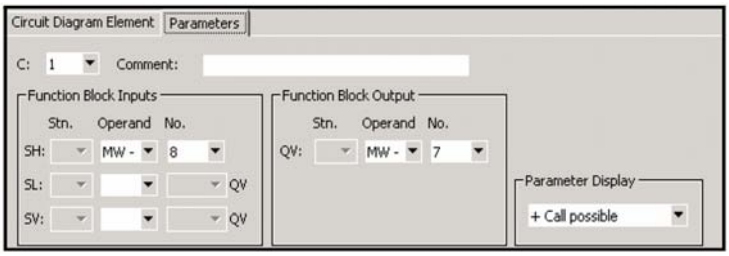
③ラダー回路作成

このプログラムのラダー回路は以下のようになっています:
サイクルマーカ M01（フリーティング接点）がカウンタ C01 に対してパルス発生装置として作動します。
パスワードが正しいかどうかの確認にはコンパレータ CP01 が使用されています。MW09 に入力されたパスワードが CP01 に設定されている値（ここでは”1234”）と“イコール”かどうか判断します。
パスワードが正しければデータファンクションブロック DB01 がトリガーされ、マーカバイト MB02 に値 3 を出力し、それがマスク 3 を呼び出します。
マスク 3 ではユーザはカウンタ C01 の設定値を変更する権限を与えられます。
また、マスク 2 で MW09 に入力されたパスワードは 3 秒後には自動で消去されるように設定します。でないとパスワードが常に入力された状態になってしまうからです。



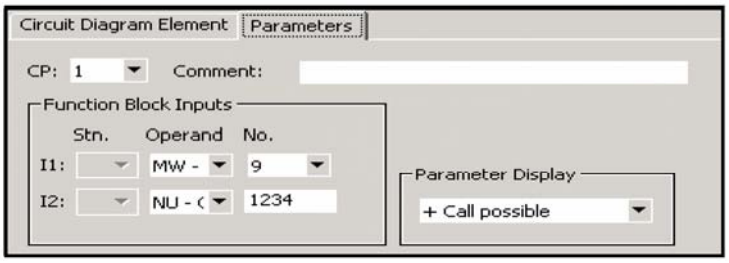
それぞれのファンクションブロックの設定

1.ファンクションブロック C01



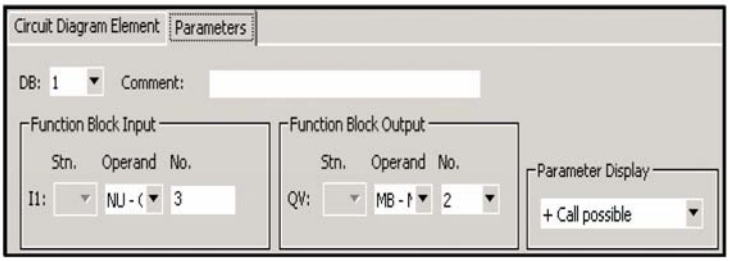
カウンタ C01：
カウンタは入力パルスをカウントし、それらを信号として MW07 の QV に出力します。上限設定値（SH）は MW08 によって決められています。

2.ファンクションブロック CP01



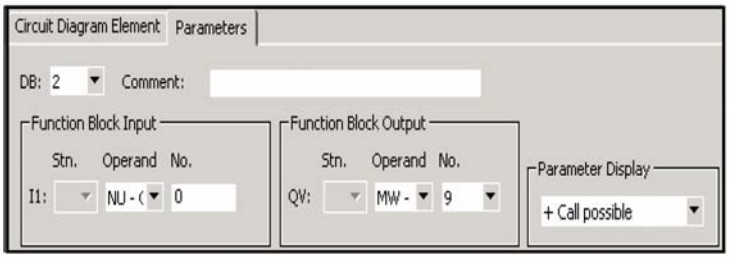
コンパレータ CP01：
これは MW09 から I1 に入力された値と入力 I2 の定数値（パスワード 1234）を比較します。EQ="Equal"

3.ファンクションブロック DB01



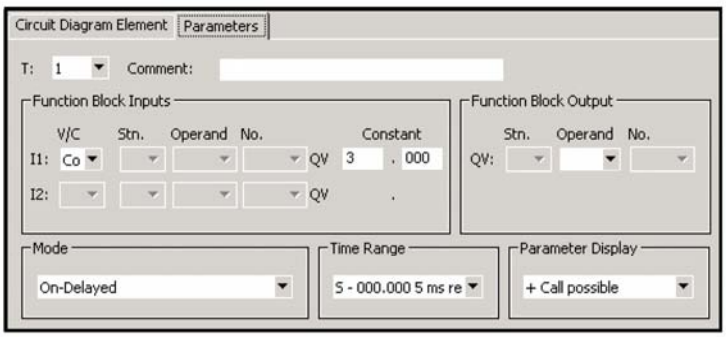
データブロック DB01：
これはトリガーされると値 3 を MB02 の QV に出力します。

4.ファンクションブロック DB02



データブロック DB02：
これはトリガーされると MW09 の QV を 0 にします。

5.ファンクションブロック T01

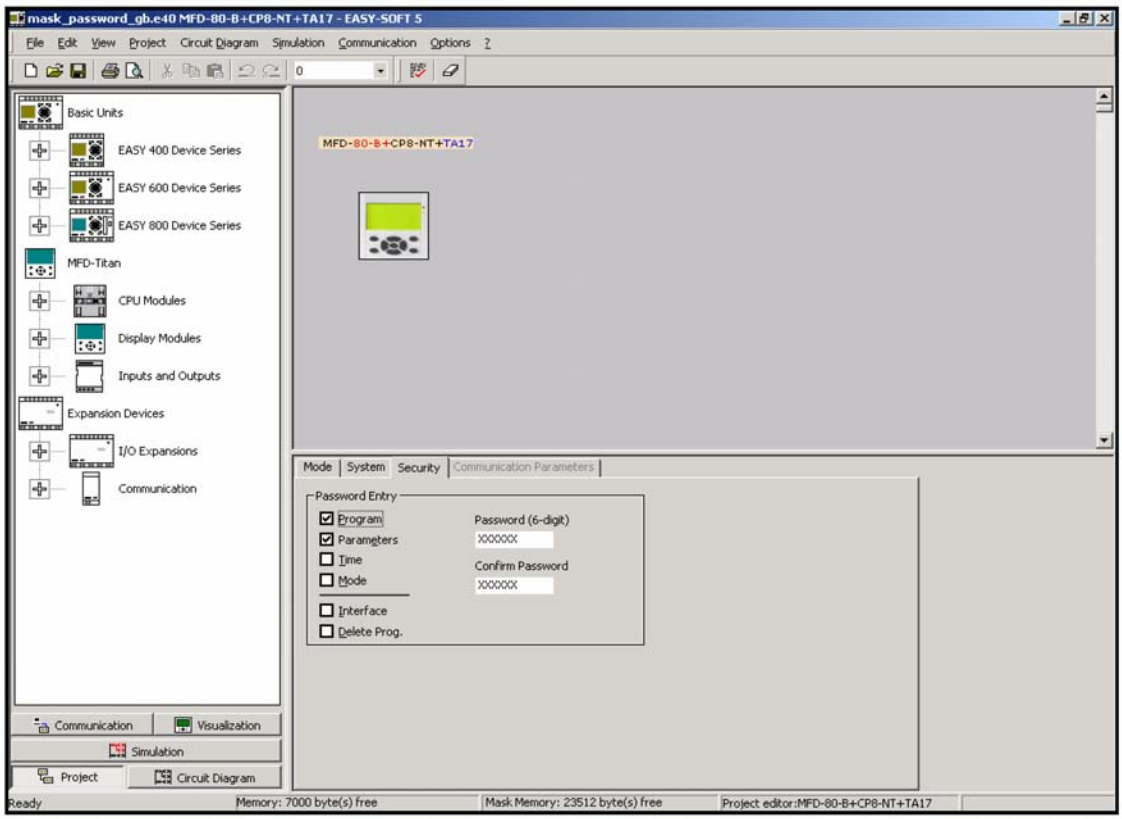


タイマ T01:
これはトリガーされると 3 秒のオンディレーで作動します。

④実際にご使用になる場合の留意点

このサンプルプログラムを実際にお客様がご使用になる場合には、Project メニューで回路図とパラメータ表示画面にお客様自身のパスワードを設定してください。コンパレータ CP01 に設定しているマスク切り替えのパスワードが保護されます。ダウンロード時にパスワードは以下のように設定されています。

プログラム、パラメータ:602000
CP01:1234



以上は MFD-Titan のビジュアルプログラムについてのご紹介でした。
なにかご不明の点がありましたら、お近くのセールスエンジニアまで、
お気軽にお尋ねください。