

ラダー図を用いた作画ソフトウェア：SHELLPA の開発

多喜康朗*¹⁾ 名和祥光*²⁾ 松本博貴*²⁾

1. はじめに

作画ソフトウェアまたは「作画ソフト」とは、プログラマブル表示器の画面データを作成するための、パソコン上で動作する専用アプリケーションソフトである。プログラマブル表示器の詳細についてはここでは割愛するが、従来のスイッチやランプなどの機能を、LCDなどの表示デバイスとマイクロコンピュータによって実現する機器である。その心臓部であるマイクロコンピュータのプログラムデータの一部をこの作画ソフトによって作成することにより、個々のユーザが望む操作表示パネルが仮想的にLCD画面上に完成する。実際のスイッチやランプを操作パネル等で使うためには配置の決定と配線作業が必要であるのと同じ様に、プログラマブル表示器の場合はそれが作画ソフトによるプログラム作業に置き替わったと考えられる。

プログラマブル表示器が商品として発展してきた歴史は、あまり明確ではないようだが、2つのルーツが考えられる。一つは記銘式集合表示灯の欠点を補うべく開発されたプログラム式の（小形）文字表示器である。もう一つは、パネルコンピュータ並みの機能を維持しつつプログラム作業を簡単にした大形グラフィック表示器（表示画面約9インチ、当社HD3形相当）である。両者ともにプログラム作業をサポートするための専用の作画ソフトが必要である。前者の作画ソフトはシンプルであるが、後者の作画ソフトは文字設定のサポートから、各種図形やグラフ描画などのグラフィックまで様々な機能をサポートする。しかも今日では、2～3年前まで大形表示器にしかなかった機能が、中形表示器（表示画面約5インチ、当社HD2L形相当）に搭載されるようになってきている。このような流れの中で、作画ソフトがサポートする機能も増加の一途をたどっていることは言うまでもない。

プログラマブル表示器と作画ソフトが高機能化する中で、作画ソフトが解決しなければならないことは、各種

の機能をいかにユーザに使い易く提供するかということである。近年は、作画ソフトが盤面機器などの図形を予め用意しておき、ユーザの手間を省く傾向にある。例えば、タッチスイッチの設定を選択すれば、タッチスイッチの図形が自動的に現れ配置される。これは作画ソフトの非常に大きな進歩の一つではあるが、系統だった進歩ではない。

コンピュータのプログラムの作成に用いるプログラミング言語に系統的な手法があるのと同様に、作画ソフトにも、統一した系統的な作画手法が存在することが望ましい。しかし現実には、一つの作画ソフトで単純な機能から複雑なものまでサポートするわけであるから、全ての機能に対して統一した作画手法があるわけではない。また業界にて標準化された手法もない。それにもまして、各社は差別化を図るべく新機能を投入しているので、新しい作画ソフトにはそれに対する新たな作画手法を提供しなければならず、作画手法という観点からはあまり収束する兆しはない。プログラマブルコントローラ（以下PCという）のプログラミングにラダー図という手法が多用されていることと比較すれば、大きな違いであり、ユーザにとっては決して歓迎すべき状況ではない。なぜならば、ユーザが異なったプログラマブル表示器を使おうと思えば、その作画ソフトの作画手法を新たに習得し、慣れる必要があるからである。そのためユーザは、機能に応じてプログラマブル表示器を選択するのではなく、一度使用した機種を機能の高低にかかわらず継続して使用する傾向がある。このようなユーザに新機種を使用してもらうには、初めての人にも分かりやすい操作性を提供しなければならない。

このような中で、“初めてのユーザにも使い易い作画ソフト”というテーマの下、HG1A形の専用作画ソフトSHELLPAを開発した。SHELLPAの一番大きな特長は、PCに広く使われているラダー図をプログラマブル表示器の世界に初めて導入したことである。それによって、PCを使い慣れているユーザには非常に馴染みやすく、動作設定状況が一目で把握でき、設定の編集（コピー、カット&ペースト）が自由になるといった長所がある。また、ラダー図を利用することによってSHELLPA独特

*¹⁾ 和泉データロジック株式会社 開発部

*²⁾ 和泉電気株式会社 商品開発部

のデバッグ機能を実現でき、動作チェックに威力を発揮する。本稿ではこれらの特長を順に解説する。

2. 作画ソフトウェアの役割

プログラマブル表示器は作画ソフトを用いたプログラムによって、個々のアプリケーションに適した運用が可能になっている。従ってユーザは、制御盤、操作盤や各種工作機械のメーカーの設計者である。

作画ソフトを使っての作業は、図1のように大きく次の3つに分けられる。

1) 作画:

プログラマブル表示器に表示される画面を具体的に作成(プログラム)する、作画ソフトの一番中心となる機能。

作成する各要素は一般的に、固定描画と動作に分けられる。固定描画とは、画面上の各種図形や文字などの、画面表示中に描画状態が更新されることのない固定的なプログラムデータである。

一方動作とは、レジスタのデータを数値として画面上に表示したり、演算などの画面表示中も内部状態の変更を伴う動的なプログラムデータである。

2) ダウンロード:

作画ソフトの作画作業で作成したプログラムデータをプログラマブル表示器に転送する作業。

SHELLPAの場合、PCとの通信プログラムも同時に転送する。

3) デバッグ:

ダウンロード終了後に、作成したプログラムが、設

計者の意図通りに正しく作られているかの検証を行う。

SHELLPAの場合、シミュレーションと運用モニタの二種類の機能を備えている。シミュレーションは当社のHD3形の作画ソフトにも既に搭載しており、画面データの検証に効果を発揮している。今回のSHELLPAでは更に運用モニタを加え、デバッグ機能の強化を図った。なお、このデバッグ時に誤りが見つかった場合、1)の作画から作業を繰り返す。

ラダー図を応用したSHELLPAでは1)の作画と3)のデバッグの作業時に効果を発揮する。市場での作画ソフトの多くは作画とダウンロードまでのサポートに留まっているが、SHELLPAではデバッグ機能も重要視している。

3. ラダー図の導入

3.1 作画手法

PCのシーケンスプログラムの手法としてラダー図が統一的に多用されていることに比較し、作画ソフトの作画手法はメーカー各社で様々である。具体的には固定描画と動作のうち、固定描画の作成手法には今日ではお馴染みのCADやお絵かきソフトの手法を取り入れている。その一方、動作の作成には、特に優れていてなおかつ標準化が進められている手法は存在しない。

動作設定手法の一つとして、作画機能をタグと呼ぶ要素に機能分割し、色々な種類のタグを設定していく手法がある。各社の作画ソフトが備えるタグ種類の例を表1に示す。各社とも呼び方と種類の分け方が異なるが、例えばタッチスイッチ、グラフ、数値表示などのタグは各社ともに備えている。固定描画はタグの種類に含まれておらず、画面上に固定される文字や図形は別に設定する。タグの設定は独立しており、一つずつ順に設定する。

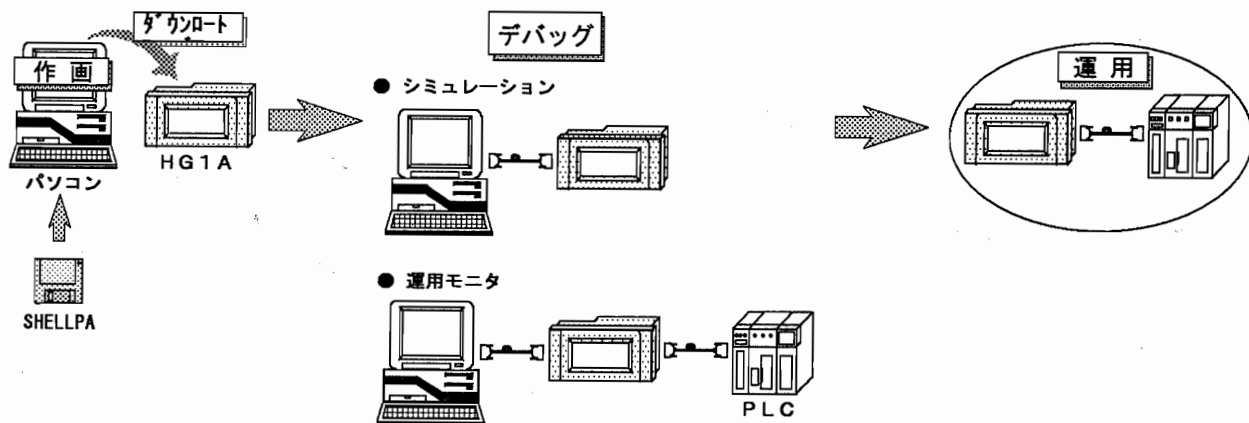


図1 SHELLPAによる作業

表1 各社作画ソフトのタグ種類の例

A社	B社	C社
時刻表示	文字タグ	画素ビット表示
図形移動表示	図形タグ	画素間接表示
グラフ表示	置換タグ	ミックス文字表示
マーク移動表示	移動タグ	ランプ表示
設定値入力	データタグ	タッチパネル
キーボード入力	バーグラフタグ	棒グラフ
ライブラリー表示	SWタグ	折れ線グラフ
マーク表示	時計タグ	数値表示
数値データ表示	テンキータグ	設定値表示
文字列表示		レール設定
タッチパネル入力 etc.		ファンクションキー

タグのもつデータは、例えば表1のライブラリー表示タグでは、タグ名、表示モード、ビットアドレス、タグの位置、等のデータである。このデータを変更したい時は、該当するタグを呼び出して再編集する。この設定と編集作業において、タグ間相互の関連は必ずしもわかりやすいとは言えない。タグは独立していることが前提条件だからである。

しかし、プログラマブル表示器に複雑な表示をさせる場合、タグの個数は多くなり、タグ間の関連も多くなって来る。そうした場合、せっかく設定した機能の確認が容易にできず、再編集の際にタグ間の関連が把握しづらいという欠点がある。

3.2 動作設定図

タグ方式による設定も含め、プログラマブル表示器の動作設定は一つのプログラミングである。プログラムの基本要素はデータ構造と制御アルゴリズムである。様々なコンピュータ言語は両者を表現しているが、プログラムの内容を理解し易いとは言い難い。設定した機能の確認や再編集を容易にするためには、設定状態を図的に表現するのが状態を視覚的に把握でき、非常に効果的な対策である。図2には、現在実用に供されていると思われる図的な表現方法を列举してみた。

コンピュータ言語に比べ、図や図的記述法ではデータ構造と制御アルゴリズム両者の表現は十分といえず、特長がどちらかに偏ることが多い。例えば、回路図は信号(データ)の処理経路を表し、フローチャートは処理手順を表すのに適している。

今日PCのプログラムとして一般に用いられているラダー図は、もともとリレー回路図である。PCではリレーの多様性を統一化して制限を加え、動作の順序、動作時間に規則性を与えた。そのためラダー図は、リレー回路図の基本構成は取り入れているが、意味合いは多少異なったものになっている。例えば、リレー回路図では順序といった概念は必要ないが、ラダー図では上から順にCPUにて処理される。これはすなわち制御手順の表現が導入されていることを示す。

制御アルゴリズムを充分に表現するために、近年ではペトリネットを元にしたSFCも導入され、ラダー図などと併用して使用されている。しかし、PCのプログラミングにラダー図が今日まで標準的に用いられ、また将来的にもそうであると思われるのは、ラダー図がプログラム言語としての適性を持っていること、つまりデータ構造と制御アルゴリズムを適度に解りやすく表現しているからであると考えられる。そこで作画ソフトの動作設定にもラダー図を導入することに着目した。

データ構造	電気回路図 論理回路図 フィードバック制御回路図 リレー回路図	コンピュータ 言語
	ラダー図	
制御 アルゴリズム	フローチャート 状態遷移図 ペトリネット(SFC)	

図2 図的なプログラム手法

3. 3 ラダー図導入の特長

作画ソフトの動作設定をラダー図にて表記することにより、次の長所がある。

- 動作設定をグラフィカルに、なおかつ集中的に表現することができる。
- 制御のフローとデータの操作を表記することで、動作設定の流れを容易に把握できる。
- 設定の変更や同じ機能設定のコピー、削除などの各種編集作業が容易にできる。
- 表示器ユーザは、PC及びラダー図に使い慣れているため、それをを用いた作画ソフトの習得も容易である。

4. ラダー図による動作設定

4. 1 SHELLPAラダー図の基本要素

PCのラダー図の基本要素は、接点、リレー出力、データ転送、演算、タイマなどである。ラダー図を作画ソフトに応用するために、プログラマブル表示器固有の機能を基本要素として追加した。SHELLPAラダー図を構成する基本要素を表2に示す。

起動とは、SHELLPAのラダー図に接続された部品や命令の起動条件であり、この起動条件が成立すると個々の部品や命令が持つ固有の機能が実行される。

部品とは、動作することによって表示画面が更新されるような要素であり、数値表示器などがこれに当たる。また命令は、動作時に表示画面に影響を与えない要素で、四則演算や論理演算がこれに類する。このことから起動は、起動条件を付与された部品または命令といえる。

図3にラダー図の一例を示す。左はタッチスイッチ、右はデータ転送を表すシンボルで、タッチスイッチが押されたときにデータ転送が行われることを示す。

数値表示やグラフなどは常にデータ表示を更新する使い方が一般的であるので、これに便利のように“常時起動”というダミーの起動条件を用意している。

4. 2 設定手順

SHELLPAが作画可能な状態の時のパソコンのモニター画面は、図4に示すように画面ウィンドウと動作ウィンドウがある。画面ウィンドウは運用時の表示画面(HG1A形)のイメージをモニタし、動作ウィンドウではラダー図を表示して動作設定を確認できる。一つの機能要素の設定に必要な手順を、タッチスイッチを例に次に示す。

- ① メニュー「起動」から「タッチスイッチ」を選ぶ。すると図5のようなタッチスイッチ設定のダイアロ

表2 SHELLPAラダー図の基本要素

起動	部品	命令
常時起動	メニュースイッチ	ベース画面切替
タッチスイッチ	テンキー	データ書込
ファンクションスイッチ	ランプ	データ転送
接点	数値表示器	文字列変換
比較接点	グラフ表示器	インクリメント
	メッセージ表示器	デクリメント
	図形表示器	四則演算
	時計	論理演算
		タイマ
		スイッチロック

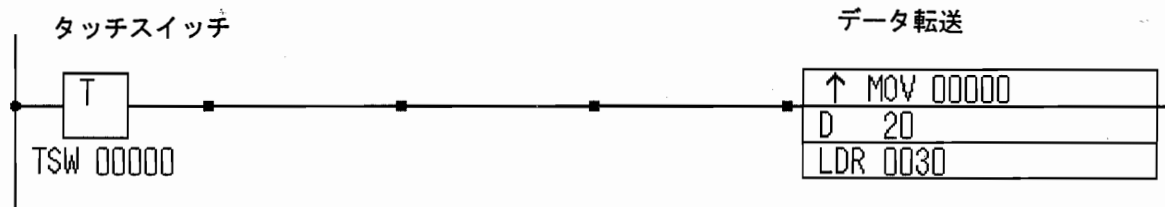


図3 ラダーシンボルの例

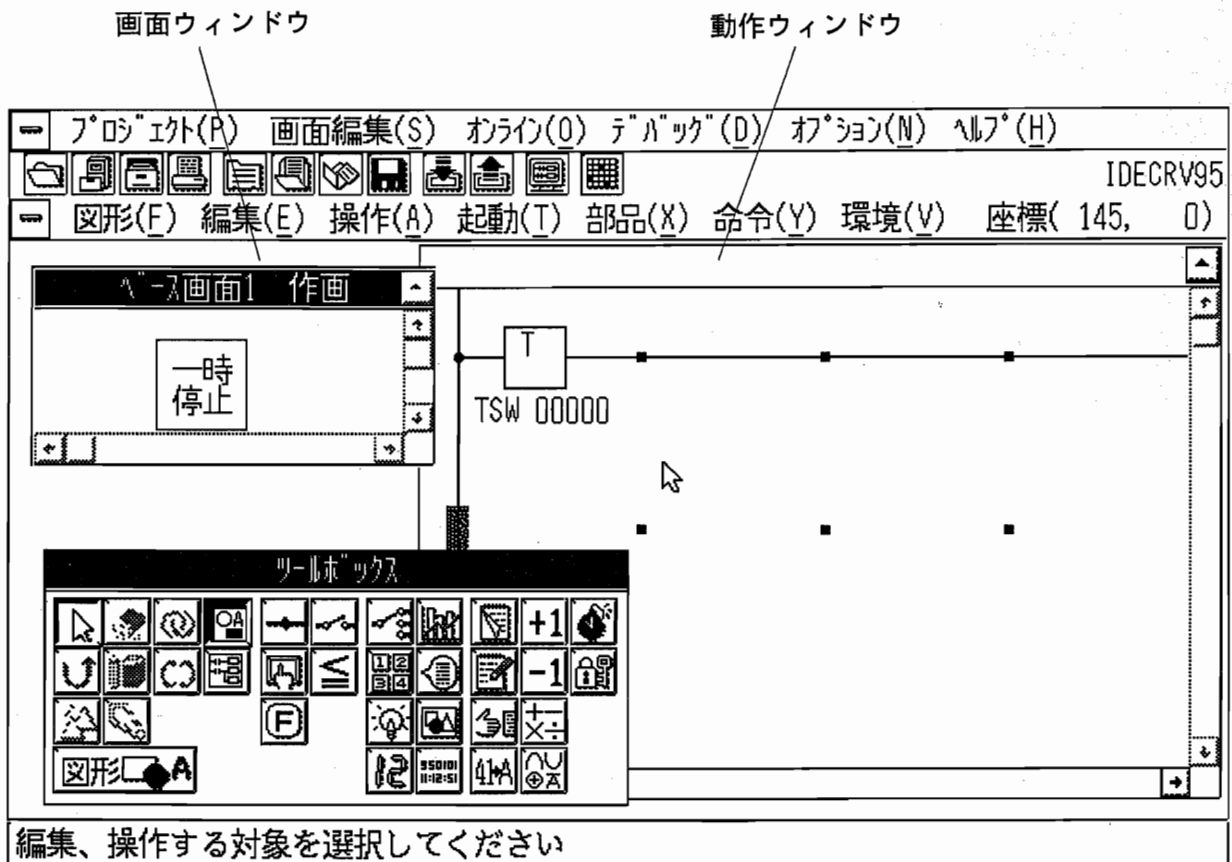


図4 作画時のSHELLPAの画面

グボックスが現れる。

- ② ダイアログボックスの必要なパラメータを設定する。このパラメータには、a接点/b接点、オルタネイト/モメンタリ、描画形状、記名文字などがある。
- ③ ダイアログを閉じると、動作ウィンドウのカーソル位置にタッチスイッチのラダーシンボルが表示される(図4の動作ウィンドウ参照)。
- ④ ③と同時に画面ウィンドウには、タッチスイッチの設定領域を示す枠が左上端に表示される。この枠をマウスのドラッグによって移動や拡大/縮小、タッチスイッチの領域の位置や大きさを決める。図4の画面ウィンドウに画面イメージを示す。

他の機能要素も基本的に同様な手順で一つずつ作成していく。ただし、タッチスイッチを除く起動と命令の全てでは、④の手順はなく、③で完了する。

②で設定するパラメータの多くは確認がラダー図上で容易にできる。例えば図3のタッチスイッチとデータ転送では、図6のように読みとれる。

4.3 応用設定

PCのラダー図にて通常設定可能な、以下の2つのラダーシーケンスもサポートしている(図7参照)。

- 二個以上の起動のAND/ORやその組み合わせができる。
- 二個以上の部品や命令の並行処理ができる。

5. デバッグ機能

5.1 シミュレーションと運用モニタ

動作設定はプログラミング作業であるため、そこには下記の二つの間違いが混入する可能性がある。

- 設計の間違い
- 作画作業時の設定の間違い

デバッグ機能は、ユーザがこれらなるべく早期に見出し、除去することを手助けするのが目的である。現在の市場には、デバッグ機能を提供する作画ソフトは少なく、SHELLPAの大きな特長となっている。またそれに加えてSHELLPAでは、ラダー図を活用し、より視覚的なモニタ機能を実現した。これにはパソコン、表示器、

タッチスイッチ

ID: TSW 00000


コメント:

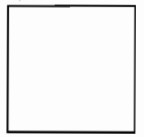
動作モード
 モノタリ 脉冲ネイト

接点方式
 A接点 B接点

表示図形なし

表示設定

表示形状
 種別: 角形押しボタ
 形状: 

表示イメージ


枠色
 フラッシュ色:
 ボタン色:

ボタンの部色
 レンズ色: ON時 レンズ色:
 ボタン色: ボタン色:

記名文字
 名称: 一時停止
 拡大率 縦: 1 横: 1
 文字色: ON時:

図5 タッチスイッチ設定時のダイアログボックス

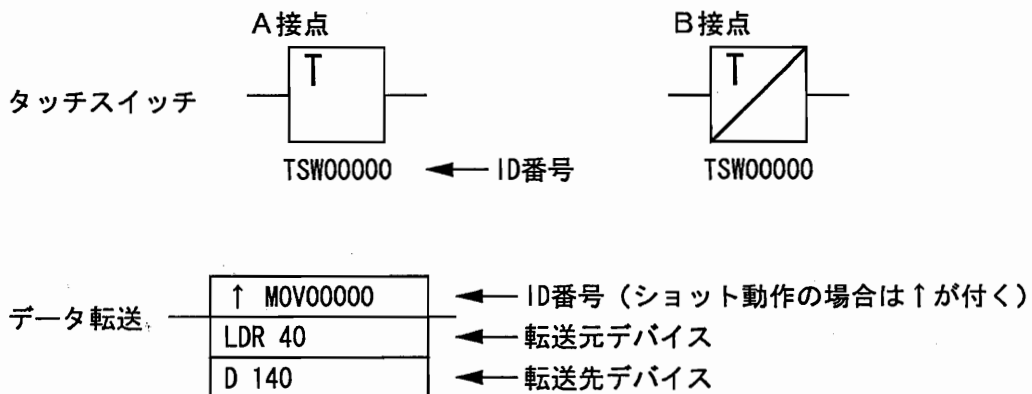


図6 ラダー図上の設定パラメータの表現例

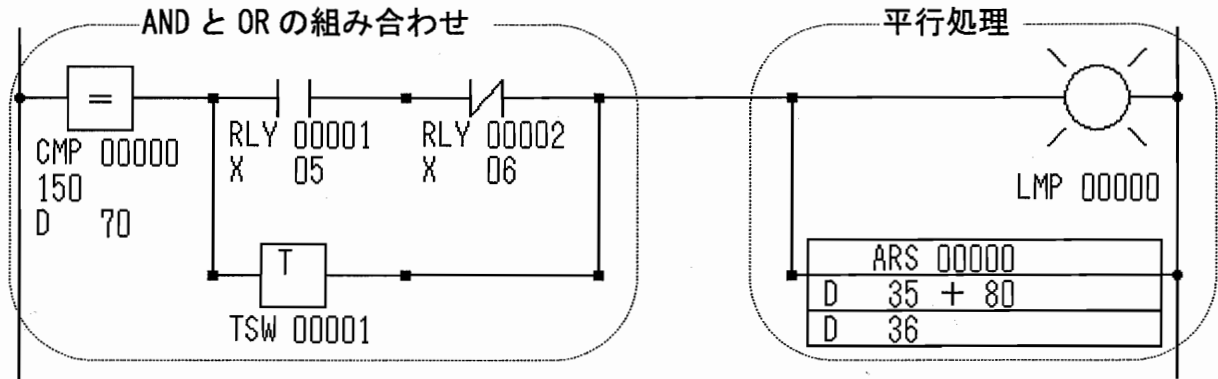


図7 ラダー図の応用設定

PCの構成によって、シミュレーションと運用モニタの2通りの方法を備えている(図1参照)。

シミュレーション：シミュレーション機能は現行HG3形と同じく、パソコンとHG1A形を接続して行う。HG1A形はパソコンがPCであるかのように通信し、パソコン(SHELLPA)はPCをエミュレートする。パソコンからのデータ入力に対するHG1A形の動きをモニターする事によって、動作設定状態を確認する。

運用モニタ：HG1A形とPC間で運用状態にある最中に、これにパソコン(SHELLPA)を接続し、運用中のHG1A形とPCの動作状態をモニターする。実際の運用状態に最も近い形で、機能チェックができる。

デバッグの実行中、パソコンの画面には現在HG1A形に表示されている画面のラダー図(動作ウィンドウ)が表示される。このラダー図上で確認できるのは、以下の二点に要約できる。

- 起動のON/OFF：ラダー図上のシンボルが状態に応じて反転表示するので、どの部品・命令が実行状態にあるか一目で把握できる。
- 設定されているデバイスのモニタ：起動・部品・命

令の設定時に登録されたデバイスの現在値が該当するシンボルの横に表示されるので、表示画面上の表示内容と比較して動作設定の確認ができる。

以上のことから設計の間違い、作業時の設定の間違いの発見が容易になる。図8にこれらの様子を示す。

5. 2 各種機能

上記の機能をより充実させるため、更に次のような各種の機能を備えている。

デバイス現在値モニタ(登録モニタウィンドウ)

任意のPCデバイス(と表示器内部デバイス)を登録すると、その現在値がパソコン(SHELLPA)の画面に表示される。運用モニタでは、HG1A形を経由してPCからデータを読み込む。これを利用すると、起動・部品・命令に登録されたデバイス以外のデバイス値をモニタできる。

データセット、リレーセット/リセット

任意のPCデバイス(と表示器内部デバイス)を登録し、そのデータを任意の値にセットすることができる。運用モニタでは、HG1A形からPCのデバイスにデータ

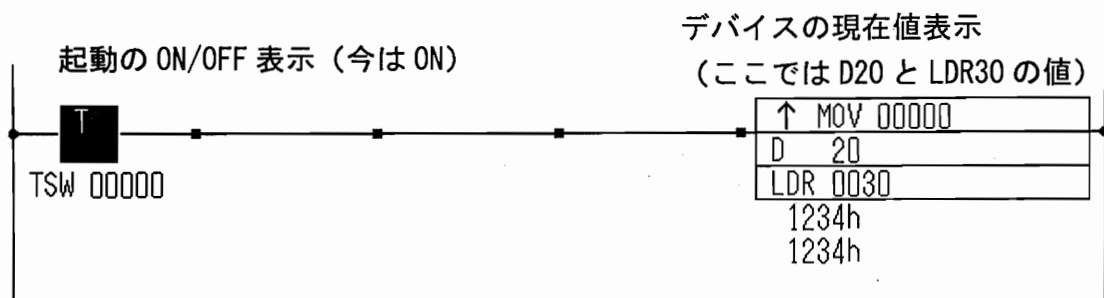


図8 デバッグ時のラダー表示

を書き込む。テストデータを入力し、その効果をHG1A形と作画ソフト（ラダー図）上で確認できる。上記デバイスの現在値モニタと併せると、余分なモニタ用の設定が不要となる。

ステップ実行

設定されている動作を、ラダーのステップ毎に一つずつ確認しながら実行させることができる。これにより、ステップに誤りがないか一つずつ確認することができる。

ブレークポイント、スキップポイント

デバッグ実行時に任意のラダーステップで実行を一時停止させたり（ブレークポイント）、任意のラダーステップを飛び越して実行させたり（スキップポイント）することができる。これにより、ラダー変更を最小限にしながらテストができる。

6. おわりに

作画ソフトのパフォーマンスは、

- 1) ユーザインタフェース
- 2) デバッグ
- 3) 作業スピード

で決定されると考えている。SHELLPAはWindowsライク（MS-DOS上で動作）な操作性を備え、多彩なデバッグを支援する。しかしそのためにソフトウェアのサイズが大きくなり、古い機種のパソコンでは実使用に多少の不満を感じるかもしれない。

Windowsの普及に伴い、ソフトウェアのユーザインタフェースは飛躍的に向上し、標準化されてきている。SHELLPAもそれに追随するべく、具体的には以下の点に注意しながらユーザインタフェースを作成した。

- (1) 文字による情報伝達よりもグラフィックな表示。
- (2) テキスト入力よりも選択肢を提供。
- (3) 対話的な操作性の実現のために、視覚的なフィードバックを提供する。例えば、その状況で利用できる選択肢を明確に示すこと。

制御盤が使用される現場は、3K（きつい、きたない、きけん）と呼ばれる業界も少なくない。プログラマブル表示器を使ったオペレータインタフェースの向上はすなわち作業の危険度を下げることである。プログラマブル表示器を使うことによって制御盤を非常に洗練されたイメージに仕上げることができる。作業（オペレータ）とのインタフェースは、作業のイメージに重要な役割を果たすと思われる。表示器を通して3Kのイメージを少しでも改善する効果があることも期待し、これからも先進のオペレータインタフェースを提供する所存である。

謝辞

作画ソフトウェアSHELLPAは、仕様・設計は和泉電気株式会社を中心に、実際の作成はアイデックシステムズ株式会社を中心に担当した、共同開発の成果である。アイデックシステムズの関係各位の熱意と努力に、この場を借りて感謝する。

参考文献

- 1) 多喜康朗, 新たなるヒューマンインタフェース, フェクトリ・オートメーション, Vol.12, No.6, 1994.
- 2) 関口隆, シーケンス制御工学, 初版, 社団法人電気学会, 1988

