DSHSIM SECS/HSMS シミュレータ

バージョン－３．０

ユーザーズ・マニュアル

２００８年6月（改-７）

株式会社データマップ

［取り扱い注意］

- この資料ならびにソフトウェアの一部または全部を無断で使用、複製することはできません。
- 本説明書に記述されている内容は予告なしで変更される可能性があります。
- Windows は米国 Microsoft Corporation の登録商標です。
- ユーザーが本 SECS/HSMS シミュレータソフトウェアの使用によって生じた遺失履歴、（株）データマップの予見の有無を問わず発生した特別損害、付随的損害、間接損害およびその他の拡大損害に対して責任を負いません。

改定履歴

番号	改訂項目	内　容
1	2005.2.14 初版	
2.	2005.3.31	ﾌﾟﾛｸﾞﾗﾑ編集に UNDO 機能を追加
3.	2006.2.16	4.9 ﾏﾆｭｱﾙｼｰｹﾝｽﾃｽﾄをｵｰﾄｼｰｹﾝｽﾃｽﾄに変更した。 送信条件、受信 MSG, ITEM 名と値を設定できるように使用を変更した。
4.	2006.3.6	(1) 通信環境定義ﾌｧｲﾙｺﾏﾝﾄﾞとして CH_LOG = <ON/OFF> を追加した。 処理履歴ﾌｧｲﾙとは別にﾁｬﾝﾈﾙ別の通信ﾒｯｾｰｼﾞ履歴ﾌｧｲﾙ記録機能を追加した。(8.1, 8.4) (2) 処理履歴画面の説明を追加した。(2.4-(2)) (3) 送受信できる最大のﾒｯｾｰｼﾞ長を 1MB まで拡張した。
5.	2006.11	(1) 処理履歴（ﾛｸﾞ）画面にﾌｫﾝﾄ、背景色設定変更のﾒﾆｭｰﾀﾌﾞを設けた。 (2) 送信ﾊﾟﾈﾙに送信 MSG ﾘｽﾄのﾌｫﾝﾄ、背景色設定変更のﾒﾆｭｰﾀﾌﾞを設けた。
6.	2007.2	(1) 4.10 ｼﾅﾘｵﾃｽﾄ機能を追加した。 ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑ言語を作り画面上でﾌﾟﾛｸﾞﾗﾐﾝｸﾞし実行できる機能である。 複数のｼﾅﾘｵを順次実行させることができる。 (2) 3.1-(4) 生成ﾌﾟﾛｼﾞｪｸﾄ情報入力画面のﾚﾎﾟｰﾄﾌｧｲﾙ関連の選択ｵﾌﾟｼｮﾝに“ﾚﾎﾟｰﾄﾌｧｲﾙを作らない”の選択肢を追加した。そしてこれをﾃﾞﾌｫﾙﾄにした。
7.	2008.6	(1) 6. ﾒｯｾｰｼﾞﾌｧｲﾙの作成と変更 ﾒｯｾｰｼﾞの削除機能（操作）を追加した。 (2) 11. 通信ﾛｸﾞﾌｧｲﾙから通信ﾒｯｾｰｼﾞ定義ﾌｧｲﾙを作成 新規に機能を追加した。

目　次

１．　はじめに

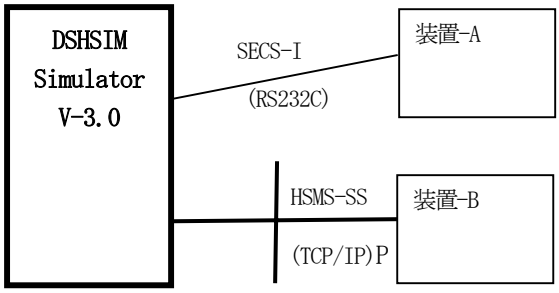
本ドキュメントは、本シミュレータを操作使用するために必要な事項について説明します。

[DSHSIM SECS/HSMS ｼﾐｭﾚｰﾀ V-3.0 ドキュメント構成]

	ﾄﾞｷｭﾒﾝﾄ名	備考
1	ユーザーズ・マニュアル　　DSHSIM User's Manual.PDF	本ﾄﾞｷｭﾒﾝﾄです。
2	プログラミング・マニュアル DSHSIM Programming Manual.PDF	

（1）DSHSIM シミュレータとは　？

DSHSIM シミュレータは、半導体製造装置工場で使用される SECS/HSMS プロトコルを使ったホスト／装置間あるいは装置／装置間通信の基本的なプロトコル機能のテストとシナリオ・シミュレーションを、パソコンを使って簡便に効率よくスピーディに行うためのソフトウェアツールです。



（2）サポートする基本的な SEMI 仕様

SEMI E4(SECS-I), E5(SECS-Ⅱ), E30(GEM), E37(HSMS), E37.1(HSMS-SS)

（3）主な機能

	機 能	備 考
1	最大８チャンネル同時通信機能	(相手装置最大８台)
2	最大通信メッセージ長	1M バイト
3	プロトコル・テスト機能 プロジェクトによるSECS-II通信実行	SECS-I, HSMS-SS 様々なエラーケースでのテスト
		自動作成プロジェクト（自動生成による）
		ユーザ作成プロジェクト（ユーザプログラミングによる）
4		マニュアルシーケンス操作によるシナリオ実行
5	通信モニタリングとログ機能	SECS-Ⅱメッセージのリスト構造形式
6	レポート機能	Excel に整理された形で表示可能
7	通信環境ファイル	チャンネル単位でプロトコルパラメータ定義可能
8	Visual オブジェクト画面の使用	シナリオ制御状態をボタン、テキストなどの Visual 画面で表現
9	プログラム機能	Visual に組み立て、マルチプログラミングで動作可能です。
10	ユーザ拡張機能	プログラムコマンドをユーザが新規に設け実装可能
11	オンラインヘルプ機能	主要な画面などの操作はヘルプ画面で即座に参照可能

（4）必要な動作環境

①) コンピュータ

No.	項目	内容
1	ＯＳ	Windows-NT 4.0 または Windows-2000, XP Windows-98(注-1)
2	ＣＰＵ	I-386 系 CPU、ｸﾛｯｸ 500MHZ 以上
3	メモリ	128Mﾊﾞｲﾄ以上 (推奨)
4	ﾃﾞｨｽｸ空き容量	200Mﾊﾞｲﾄ以上 (推奨)
5	通信ｶｰﾄﾞ	RS232Cﾎﾟｰﾄ(SECS-1 通信する場合に必要) ﾈｯﾄﾜｰｸｶｰﾄﾞ(HSMS 通信する場合に必要)
6	ディスプレイ	画像の領域： 1024 x 768 以上 画面の色 ： 16ﾋﾞｯﾄ(65,536 色)以上 (推奨) (8ﾋﾞｯﾄの場合、一部中間色の色が正しく表示されません。)
7	HASP Adapter 用	USB (Windows-98,2000,XP 用) 25Pin DSub ｺﾈｸﾀ PIO(Windows-NT-4.0 用)

(注) Windows-98 では、複数チャンネルの動作中に長いメッセージの通信時、応答性が若干悪くなるか、停止する可能性があります。その場合通信ﾛｸﾞを処理履歴画面に表示しないようにして動作させると正常になることがあります。

③ HASP アダプタ (ﾌﾟﾛﾃｸﾄﾃﾞﾊﾞｲｽ)
USB または 25Pin DSub PIO ｲﾝﾀﾌｪｰｽ用　　(注) 試用版には必要ありません。

（5）必要なファイル

No.	内 容	ﾌﾟﾛｸﾞﾗﾑ/ﾌｧｲﾙ名
1	DSH SECS/HSMS ｼﾐｭﾚｰﾀ実行ﾌﾟﾛｸﾞﾗﾑ	dshsim.exe
2	DLL(Dynamic Link Library)ﾌﾟﾛｸﾞﾗﾑ	(1) dsh_gdll.dll (2) x3dll.dll (3) v3dll.dll (4) P3Dll.dll (5) c3dll.dll
3	SECS/HSMS 通信環境ﾌｧｲﾙ	COMM.DEF (ﾌｧｲﾙ名は COMM.DEF でなくてもよい)
4	ｺﾏﾝﾄﾞ、変数定義ﾌｧｲﾙその他	(1) command.txt (2) property.txt (3) constant.txt (4) variable.txt (5) msgvar.txt
5	ﾕｰｻﾞ定義のﾒｯｾｰｼﾞ定義ﾌｧｲﾙ	message file (参考ﾌｧｲﾙ) Sample.msg, gem_s01.msg
6	ﾕｰｻﾞﾌﾟﾛｸﾞﾗﾑ生成用雛型ﾌｧｲﾙ	main_pro.org recv_pro.org pro_rep.org
7	ﾍﾙﾌﾟﾌｧｲﾙ	DSHSIM_V3.CHM

２．　シミュレータのインストールと実行開始

２．１　シミュレータのインストール

（１）　インストールの手順
SECS/HSMS シミュレータＣＤのルート ￥ の Setup.exe プログラムを起動します。
起動されると、インストール先を問合せてきますので、そこでドライブ名とフォルダー名を指定して下さい。この後、自動的にインストールされます。

（２）　インストールで保存されるファイル
１．４－（３）の表のファイルが、それぞれ指定されたフォルダーに保存されます。

２．２　ＨＡＳＰキーのインストール

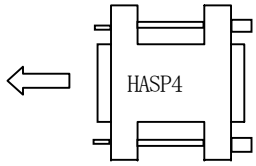
シミュレータプログラム DSHSIM.EXE 実行の前にハードプロテクトデバイス HASP キーの接続とドライバープログラムのインストールを行ってください。

（評価版プログラムについてはＨＡＳＰキーのインストールは必要ありません。）

HASP キーアダプターには①２５ピンパラレル用、②ＵＳＢ用の２種類のものがあります。

［２５ピンパラレル用アダプター］

（１）　同梱の HASP キーアダプター（25 ピンパラレルポート用）をパソコンのパラレルコネクターに接続します。

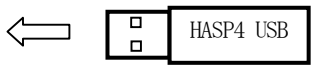


（２）　次に、１．１でインストールされた表 1-1 の 8. の **HASP.BAT** バッチファイルを実行します。
これで、HASP ドライバーがインストールされます。

［ＵＳＢ用アダプター］

（１）　**ＵＳＢ用アダプター接続しないで**、１．１でインストールされた **HASP.BAT** バッチファイルを実行します。

（２）　次に同梱の HASP キーアダプター（ＵＳＢ用）をパソコンのＵＳＢコネクターに接続します。



（３）　そのあと、システムを**一旦終了させて、再起動**をして下さい。

以上でＨＡＳＰキーのインストールは完了です。

HASP のインストールは１度行えば、システムの再インストールを行わない限り新たに行う必要はありません。

２．３　準備作業とシミュレータの起動

実際の通信を行うためには、通信環境定義ファイル（以下、環境ファイルとも呼びます。）を実際の装置の通信環境に合わせる必要があります。環境ファイルは通常、COMM.DEF の名前のテキストファイルであり、"メモ帳"などのエディタープログラムで内容を変更することができます。もちろんシミュレータ上でも変更できます。

環境ファイルのサンプルファイルは「付録－A」に示すようになっています。

変更必要な主な項目は、次のような項目です。

① ﾃﾞﾊﾞｲｽ ID　② ｼﾘｱﾙﾎﾟｰﾄ　③ﾎﾞｰﾚｰﾄ　④ HSMS IP, Socket Port などです。

シミュレータの起動は、１．１でインストールした　DSHSIM.EXE を通常の Windows プログラムの起動と同様に行います。起動パラメータとして、環境ファイル名を指定することができます。

DSHSIM　[<通信環境定義ファイル名>]

起動ﾊﾟﾗﾒｰﾀ省略時は環境ﾌｧｲﾙとしてﾃﾞﾌｫﾙﾄの COMM.DEF ﾌｧｲﾙを使用します。

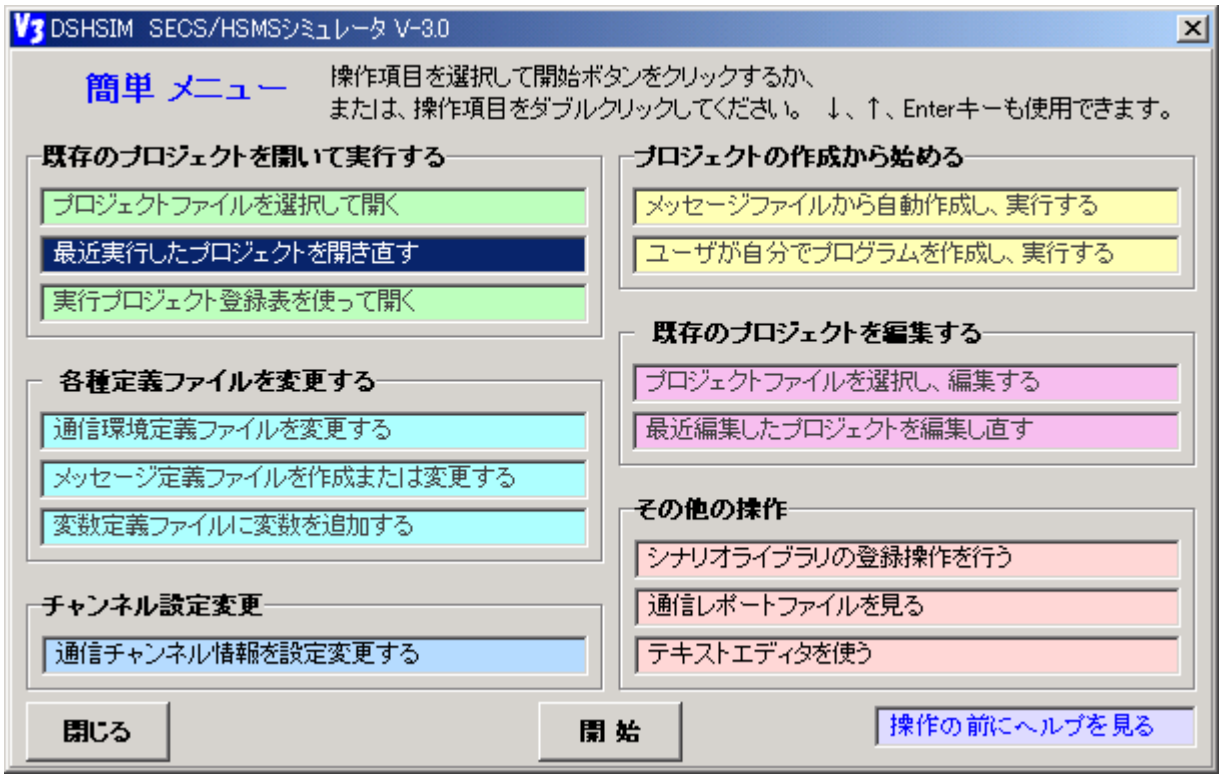
シミュレータが起動されると**メイン画面**と**簡単メニュー**が表示されます。（メイン画面は背後に隠れています。）

簡単メニューには、本シミュレータが提供する操作機能の主要なものが全て入っています。

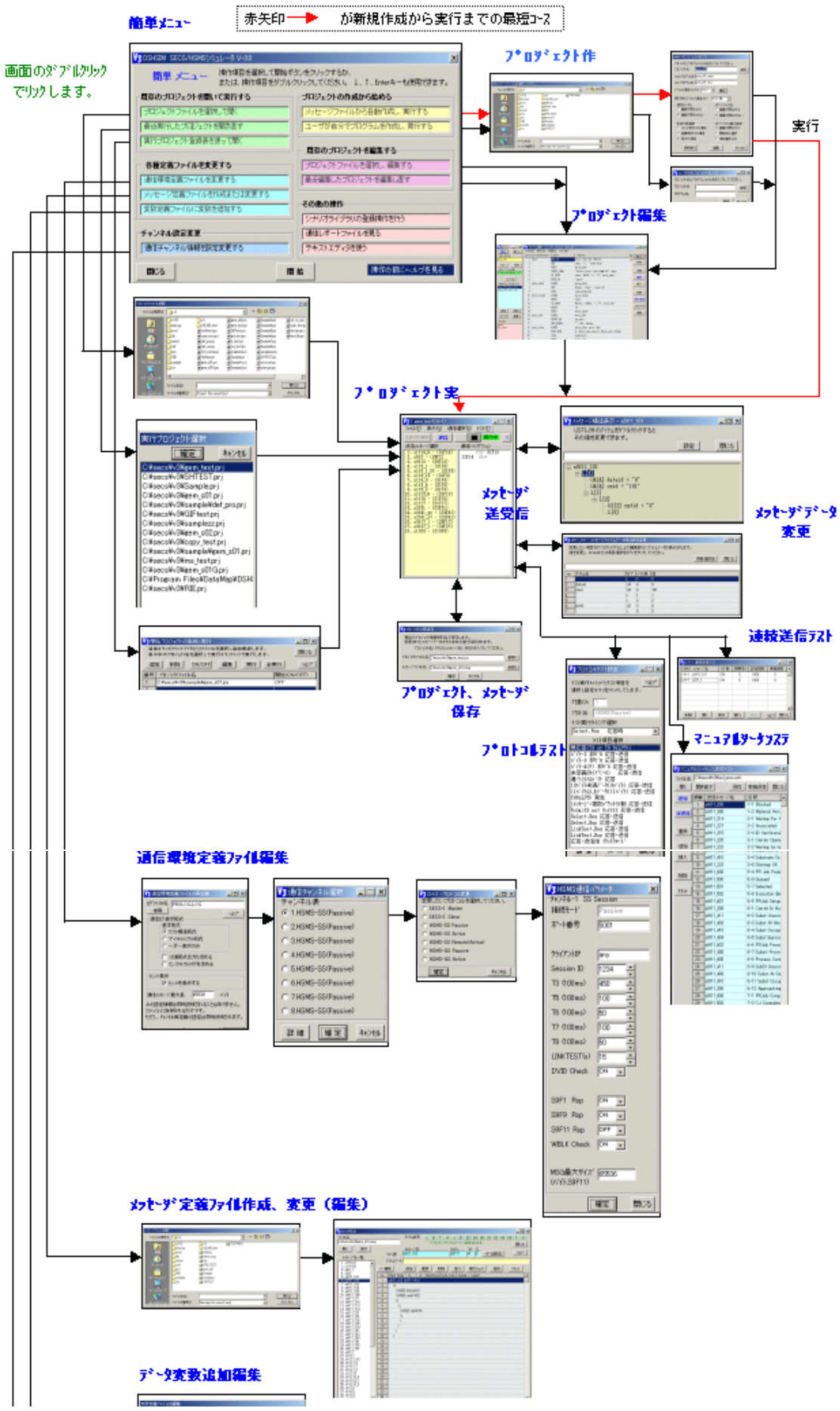
操作実行したいメニューをダブルクリックすれば、その操作がただちに開始されます。

操作を前にヘルプを見る　を実行すると、シミュレータ全体の操作と流れが分かる操作マップが表示されます。
操作マップ画面は次ページのとおりです。（ﾍﾙﾌﾟ メニューのクリックで最初に操作マップが表示されます。）

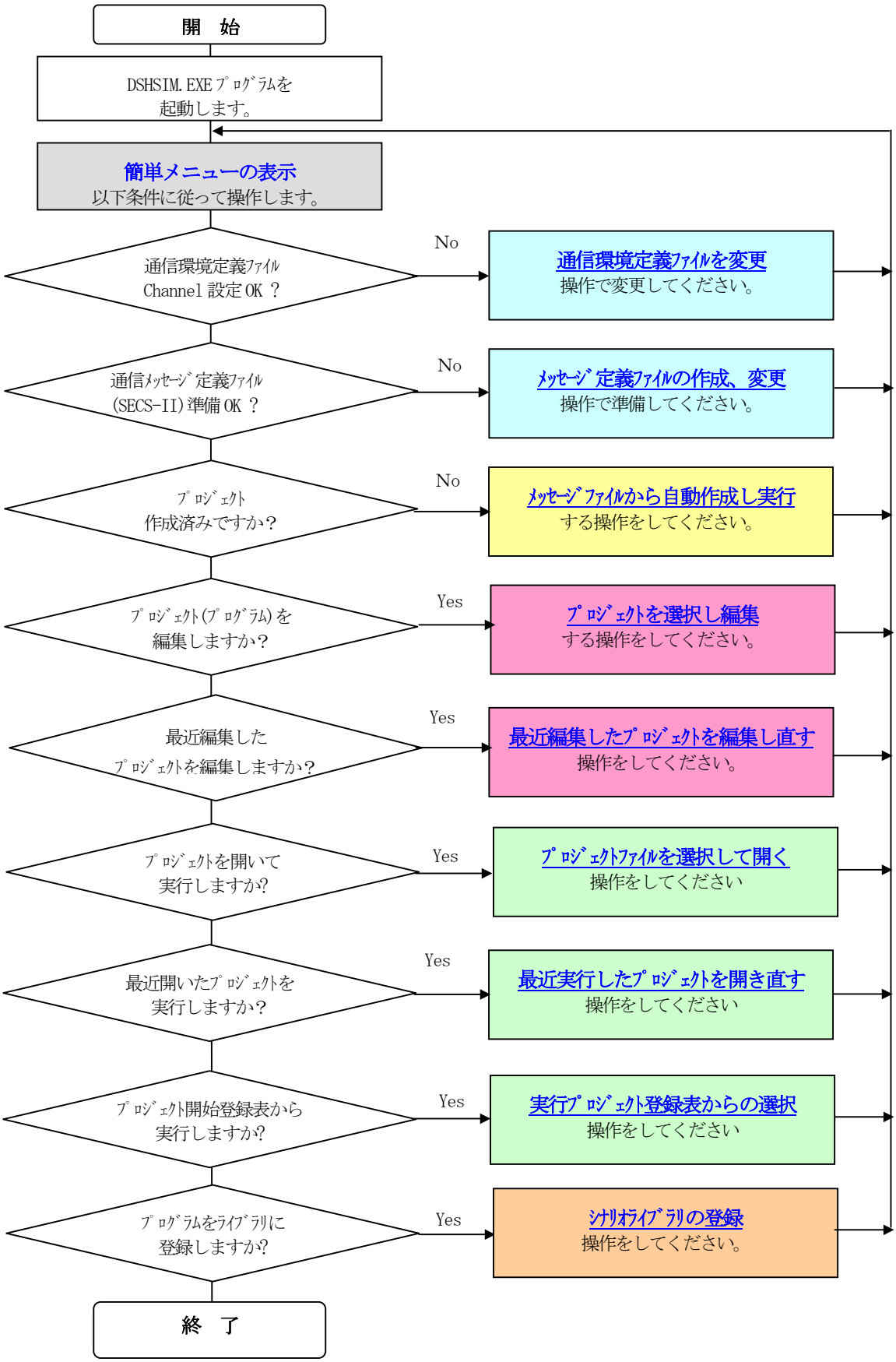
（1）簡単メニュー



（2）操作マップ


簡単メニュー
赤矢印 → が新規作成から実行までの最短コース
画面のﾀﾞﾌﾞﾙｸﾘｯｸ
でﾘｯｸします。
簡単 メニュー
ﾌﾟﾛｼﾞｪｸﾄ作
実行
ﾌﾟﾛｼﾞｪｸﾄ編集
ﾌﾟﾛｼﾞｪｸﾄ実
ﾒｯｾｰｼﾞ
送受信
ﾒｯｾｰｼﾞﾃﾞｰﾀ
変更
連続送信ﾃｽﾄ
ﾌﾟﾛｼﾞｪｸﾄ、ﾒｯｾｰｼﾞ
保存
ﾌﾟﾛﾄｺﾙﾃｽﾄ
ﾏﾆｭｱﾙﾒｰﾝﾃﾅﾝｽ
通信環境定義ファイル編集
ﾒｯｾｰｼﾞ定義ファイル作成、変更（編集）
ﾃﾞｰﾀ変数追加編集

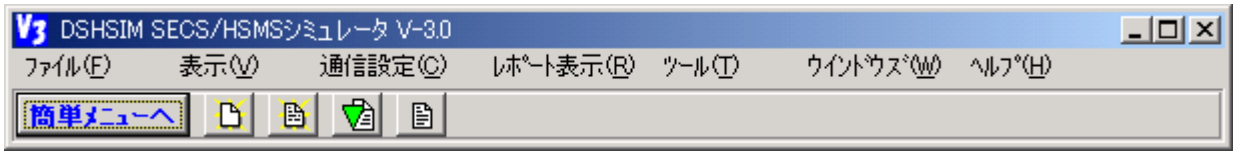
（3）操作メニューと操作フロー



２．４　メイン画面と処理履歴画面

シミュレータのメイン画面と処理履歴画面について説明します。

（1）メイン画面



① ﾌｧｲﾙ(F) メニュー

プロジェクトの実行、編集その他ファイル編集操作用メニューです。

メニュータブ	ﾘﾝｸ　／　説明
ファイル(F)	
実行ﾌﾟﾛｼﾞｪｸﾄを開く(O)	実行ﾌﾟﾛｼﾞｪｸﾄを開く(O)
実行ﾌﾟﾛｼﾞｪｸﾄを開き直す(R)	実行ﾌﾟﾛｼﾞｪｸﾄを開き直す(R)
開始登録ﾌﾟﾛｼﾞｪｸﾄを開く(A)	開始登録ﾌﾟﾛｼﾞｪｸﾄを開く(A)
ﾌﾟﾛｼﾞｪｸﾄを新規生成する(C)	ﾌﾟﾛｼﾞｪｸﾄを新規生成する(C)
ﾌﾟﾛｼﾞｪｸﾄを編集する(E)	ﾌﾟﾛｼﾞｪｸﾄを編集する(E)，　ﾌﾟﾛｸﾞﾗﾑの編集
ﾌﾟﾛｼﾞｪｸﾄを再編集する(B)	ﾌﾟﾛｼﾞｪｸﾄを再編集する(B)，　ﾌﾟﾛｸﾞﾗﾑの編集
全実行ﾌﾟﾛｼﾞｪｸﾄを閉じる(Q)	全実行ﾌﾟﾛｼﾞｪｸﾄを閉じる(Q) – 実行中のﾌﾟﾛｼﾞｪｸﾄを全て終了させます。
ﾒｯｾｰｼﾞ定義ﾌｧｲﾙを編集する(M)	ﾒｯｾｰｼﾞ定義ﾌｧｲﾙを編集する(M)
変数定義ﾌｧｲﾙを編集する(V)	変数定義ﾌｧｲﾙを編集する(V)
簡易ﾃｷｽﾄｴﾃﾞｨﾀ(Z)	簡易ﾃｷｽﾄｴﾃﾞｨﾀ(Z)　– ﾃｷｽﾄｴﾃﾞｨﾀｰを開きます。
ｼﾐｭﾚｰﾀを終了する(X)	ｼﾐｭﾚｰﾀﾌﾟﾛｸﾞﾗﾑを終了します。

② 表示(V) メニュー

メニュータブ	ﾘﾝｸ　／　説明
表示(V)	
画面を1番上に表示(T)	ｼｽﾃﾑ全体の画面表示制御の操作
●ﾛｸﾞ表示ON/OFF(O) ●ﾋﾝﾄ表示ON/OFF(N)	通信ﾒｯｾｰｼﾞ表示などについて設定します。
●全CH通信履歴ﾛｸﾞ表示ON/OFF(C)	
●全CH通信ﾛｸﾞ形式-ﾘｽﾄ構造(L) 全CH通信ﾛｸﾞ形式-ｱｲﾃﾑ形式(I) 全CH通信ﾛｸﾞ形式-ﾍｯﾀﾞｰのみ表示(R) 全CH通信履歴 16進形式表示(H)	
未定義MSGﾛｸﾞ表示ON/OFF(U)	
HSMSﾘﾝｸﾃｽﾄMSG表示ON/OFF(G)	

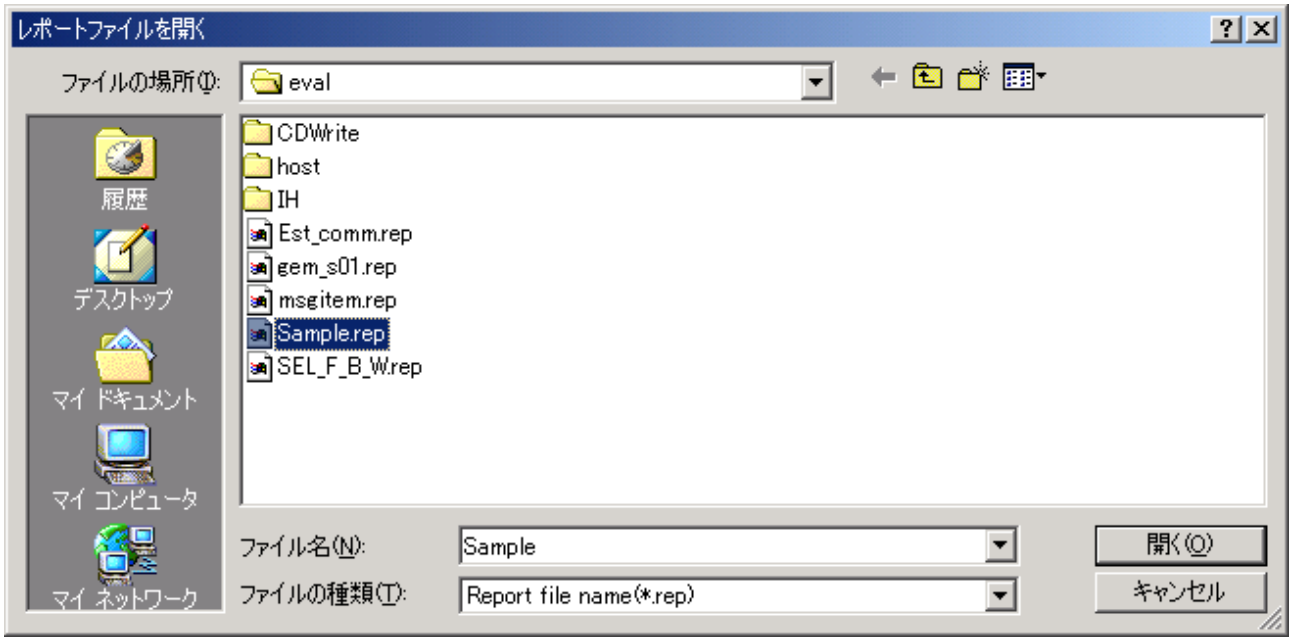
③ 通信設定(C)メニュー

メニュータブ	ﾘﾝｸ ／ 説明
通信設定(C) 通信条件の参照/設定(P) 通信状態をみる(S) 通信環境定義ﾌｧｲﾙの再定義(B) ●ﾘﾝｸﾃｽﾄ実行ON/OFF(L) S9Fx通信制御ON/OFF(F)	通信条件の参照/設定(P)
	通信状態をみる(S)
	通信環境定義ﾌｧｲﾙの再定義(B)
	ﾘﾝｸﾃｽﾄ実行 ON/OFF(L) ﾁｪｯｸ指定で、全ﾁｬﾝﾈﾙに対してﾘﾝｸﾃｽﾄ実行するかどうかを指定します。
	S9FX 通信制御 ON/OFF ﾁｪｯｸ指定で、全ﾁｬﾝﾈﾙに対して S9Fx 送信条件時、S9Fx を送信するかどうかを指定します。

④ ﾚﾎﾟｰﾄ表示(R)メニュー

レポートファイルを開いてみることができます。

最初にレポートファイルを開くダイアログ画面が表示されますので、そこで、表示したいレポートファイル名を選択します。



レポート画面例を見る。

⑤ ﾂｰﾙ(T)メニュー

メニュータブ	ﾘﾝｸ ／ 説明
ﾂｰﾙ(T) ﾌﾟﾛｸﾞﾗﾑのｼﾅﾘｵﾗｲﾌﾞﾗﾘ登録(S)	ﾌﾟﾛｸﾞﾗﾑのｼｱﾘｵﾗｲﾌﾞﾗﾘ登録(S)

⑥ ｳｲﾝﾄﾞｳｽﾞ (W) メニュー

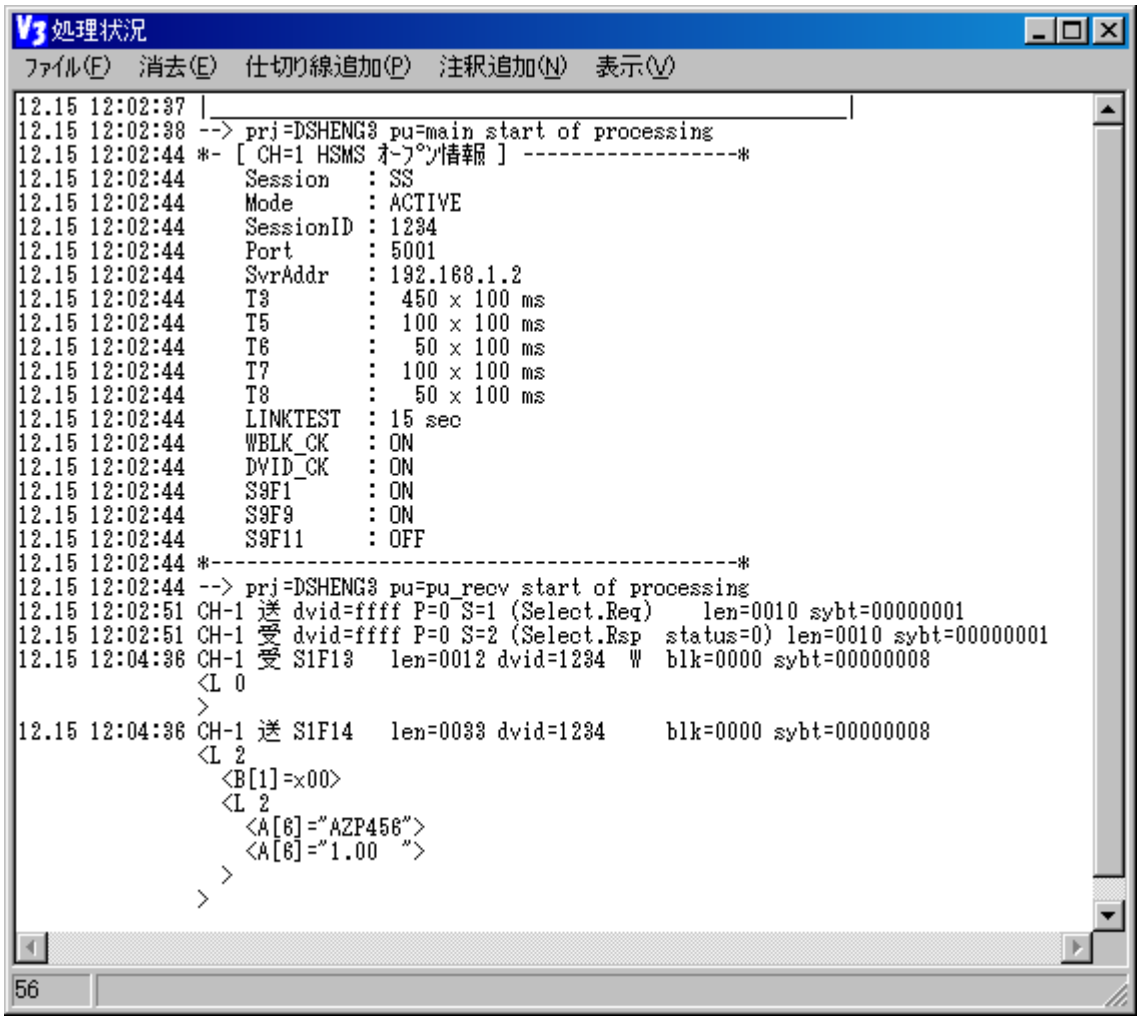
メニュータブ	ﾘﾝｸ ／ 説明
ｳｲﾝﾄﾞｳｽﾞ(W) ﾊﾟﾈﾙ-1 1. sel_host(1) 操作ｺﾝｿｰﾙ画面(O) ﾌﾟﾛｼﾞｪｸﾄ編集(R) ﾌﾟﾛﾊﾟﾃｨ設定(P) ﾌﾟﾛｸﾞﾗﾑ編集：H_MAIN.PRO（sel_host）(A) 処理履歴(LOG)(L)	現在ｽｸﾘｰﾝ上に表示されているｼﾐｭﾚｰﾀ関連の画面名が表示されます。 選択してｸﾘｯｸすると、その画面がﾌｫｰｶｽ（操作対象の画面になる）されます。 例えば､小さい画面が大きい画面に隠れてしまった場合などに使用します。

⑦ ﾍﾙﾌﾟ (H) メニュー

メニュータブ	ﾘﾝｸ ／ 説明
ﾍﾙﾌﾟ(H) ﾍﾙﾌﾟ(H) ﾊﾞｰｼﾞｮﾝ情報(V)	ﾍﾙﾌﾟ (H) – ﾍﾙﾌﾟ 画面を表示します。
	ﾊﾞｰｼﾞｮﾝ情報(V) – ｼﾐｭﾚｰﾀのﾊﾞｰｼﾞｮﾝ情報を表示します。 

（2）処理履歴表示画面(通信ログ画面)

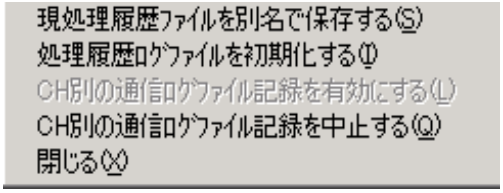
処理履歴画面はシミュレータの処理状況をモニタリングするための画面です。
送受信した通信メッセージのモニタリング表示も行います。この画面に表示された情報はログファイルに記録されます。



操作メニューとその機能は以下のとおりです。

① ﾌｧｲﾙ(F) メニュー

ログファイルのコントロールを行います。



メニュータブ	説明
現処理履歴ﾌｧｲﾙを別名で保存する(S)	現処理履歴ﾌｧｲﾙを別名で保存し、現処理履歴ﾌｧｲﾙを空にします。
処理履歴ﾌｧｲﾙを初期化する(I)	処理ﾘﾚｷﾌｧｲﾙを空にします。
CH 別の通信ﾌｧｲﾙﾛｸﾞﾌｧｲﾙ記録を有効にする(L)	ﾁｬﾝﾈﾙ別の通信履歴ﾌｧｲﾙの記録を開始します。
CH 別の通信ﾌｧｲﾙﾛｸﾞﾌｧｲﾙ記録を中止する(Q)	ﾁｬﾝﾈﾙ別の通信履歴ﾌｧｲﾙの記録を中止します。
閉じる(X)	処理履歴画面を閉じます。

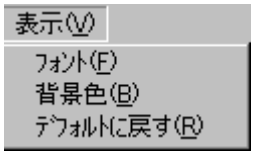
チャンネル別の通信ログファイル名は次のように予約されています。
ch?_log.log　－？の部分がチャンネル番号になります。
チャンネル別通信ログファイルには相手装置との通信メッセージの履歴だけがリスト構造形式で記録されます。

(参考)
DSHSIM では以下のログファイルを記録保存することができます。
・処理履歴ファイル ：
シミュレータの処理関連ログを記録します。通信メッセージ記録もします。
・チャンネル別の通信ログファイル（CH-1,2,3,4,5,6,7,8）
行われた通信メッセージ履歴をチャンネル別に記録します。

② 消去(E)、仕切り線追加(P)、注釈追加(N)メニュー
クリックするだけでそれぞれ以下の機能を実行することができます。

メニュータブ	説明
消去(E)	画面に表示されている情報を消去します。
仕切り線追加(P)	画面と処理履歴ﾌｧｲﾙに仕切り線を挿入します。
注釈追加(N)	画面と処理履歴ﾌｧｲﾙに注釈行を挿入します。

③ 表示(V)メニュー
画面のフォント（フォント名、色、サイズなど）ならびに背景色を設定変更できます。また、初期状態に戻すことができます。



メニュータブ	説明
ﾌｫﾝﾄ(F)	文字ﾌｫﾝﾄの変更ができます。(ﾌｫﾝﾄ名、ｻｲｽﾞ、色、ｽﾀｲﾙ)
背景色(B)	画面の背景色を変更できます。
ﾃﾞﾌｫﾙﾄに戻す(R)	ﾌｫﾝﾄ、背景色、画面位置ならびに画面ｻｲｽﾞを初期状態に戻します。

(注) DSHSIM は本画面設定時に設定情報を simlog.cnf ファイルに記憶します。
そして、起動時に simlog.cnf の内容で画面設定を行います。

３．　プロジェクトの生成

３．１　プロジェクトの新規生成操作

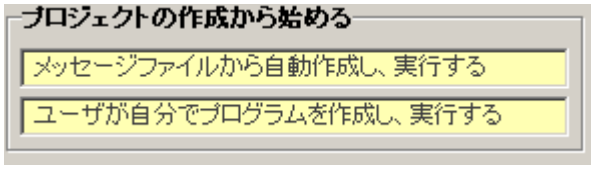
プロジェクトの新規生成を行うためには、あらかじめ、相手装置とのSECS-II通信メッセージ定義ファイルが必要です。(通信メッセージ定義ファイルのファイル拡張子は .MSG　です。)

通信メッセージファイルの作成、編集については　「通信メッセージ定義ファイル仕様」を参照してください。

シミュレータの画面上の操作から始めます。

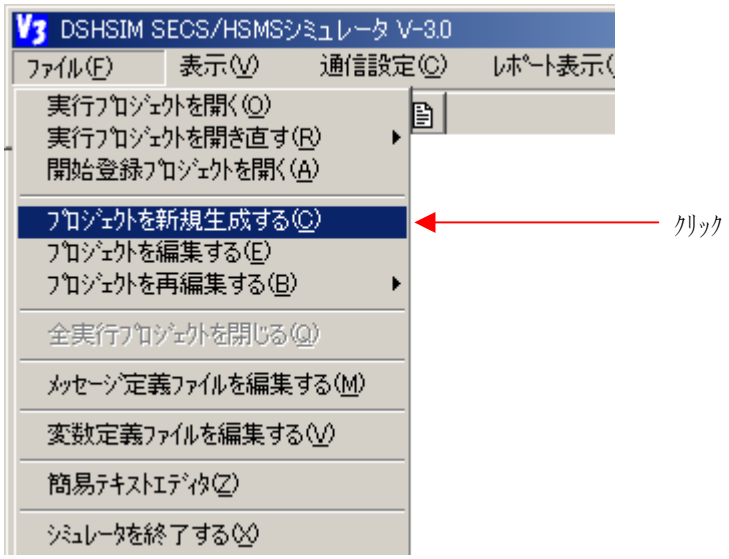
（1）生成操作の開始

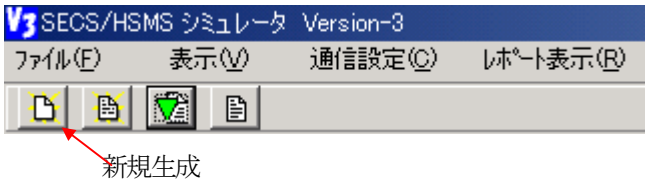
① 簡単メニューから



(簡単メニューからの操作では、次の操作はスキップされ、（３）に進みます。)

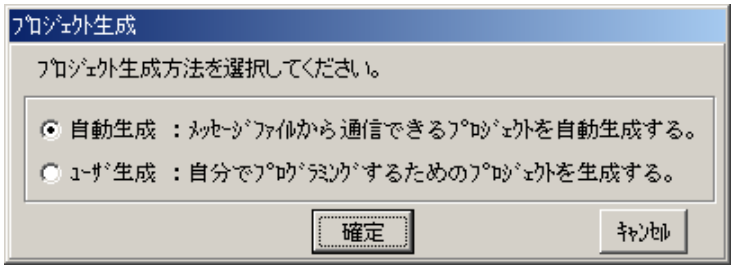
② ﾌｧｲﾙ(F)の「ﾌﾟﾛｼﾞｪｸﾄを新規生成する(C)」またはﾂｰﾙﾎﾞﾀﾝ  （新規生成）のｸﾘｯｸからはじめます。





③ 次のプロジェクト生成方法選択画面が表示されます。

（2）プロジェクト生成の生成方法選択画面



生成方法には、画面のように、２つの方法があります。どちらかを選択し、確定ﾎﾞﾀﾝをｸﾘｯｸしてください。
ｷｬﾝｾﾙﾎﾞﾀﾝをｸﾘｯｸすると新規生成の操作を中止します。

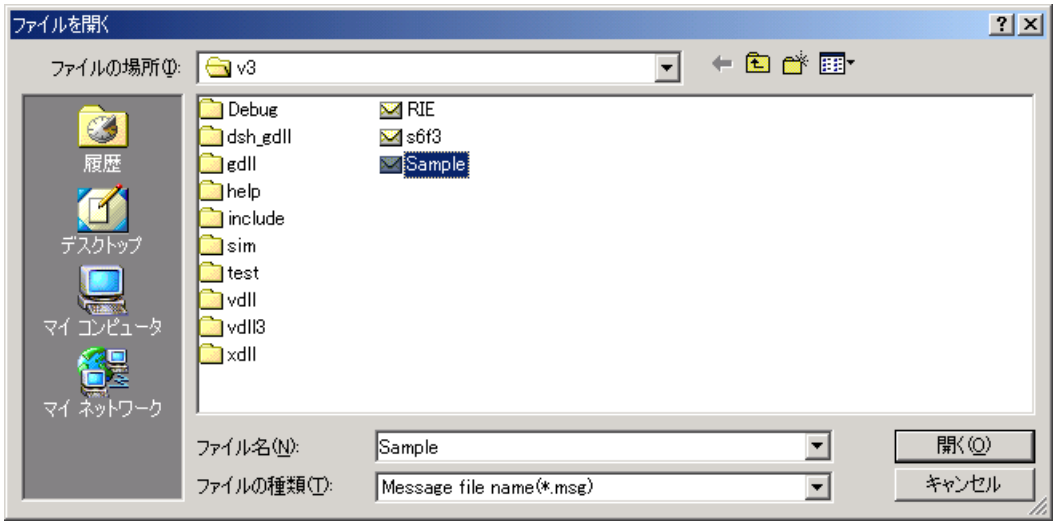
① ﾒｯｾｰｼﾞﾌｧｲﾙから通信できるﾌﾟﾛｼﾞｪｸﾄを自動生成する。（以下、**自動生成**と呼びます。）
ｼﾐｭﾚｰﾀが後での操作で指定するﾒｯｾｰｼﾞ定義ﾌｧｲﾙに定義されているﾒｯｾｰｼﾞに対する送信、受信処理ﾌﾟﾛｸﾞﾗﾑを全て自動的に生成させる方法です。
main と recv の２つのﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙが生成されます。
この方法で作成されたﾌﾟﾛｸﾞﾗﾑで基本的はｼﾐｭﾚｰｼｮﾝを行うことができます。

② ﾕｰｻﾞが自分で独自のﾌﾟﾛｸﾞﾗﾑを作成するための方法です。（以下、**ﾕｰｻﾞ生成**と呼びます。）
ﾕｰｻﾞが必要なﾌﾟﾛｸﾞﾗﾑを作成することができます。例えば、ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑ（ﾗｲﾌﾞﾗﾘ）作成時にこの方法を選択します。

③ 確定の後、通信対象装置との間で使用される SECS ﾒｯｾｰｼﾞの定義ﾌｧｲﾙ選択画面が表示されます。

（3）ＳＥＣＳメッセージ定義ファイル選択画面

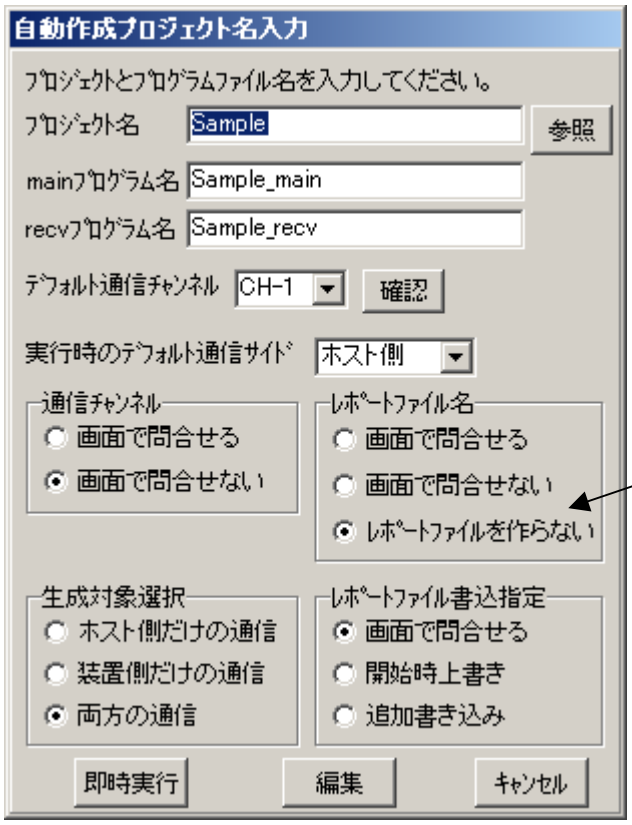
ここで、作りたいﾌﾟﾛｼﾞｪｸﾄのﾒｯｾｰｼﾞ定義ﾌｧｲﾙを選択します。
例として、ここで、Sample.msg を選択し、開くをｸﾘｯｸします。



次に、自動作成、ユーザ作成それぞれのプロジェクト名などの入力画面が表示されます。

（4）生成プロジェクト情報入力画面

【自動生成の場合】



2007 年 2 月
◎ ”ﾚﾎﾟｰﾄﾌｧｲﾙを作らない”の選択肢を追加しました。
”ﾚﾎﾟｰﾄﾌｧｲﾙを作らない”が選択されたら実効開始時にﾚﾎﾟｰﾄ関連の問合せ画面が表示されません。

① ﾌﾟﾛｼﾞｪｸﾄ名

ﾃﾞﾌｧﾙﾄで、ﾒｯｾｰｼﾞﾌｧｲﾙ名が表示されますが、名付けたい名前を入力してください。

② main ﾌｧｲﾙ名

main の PU プログラム用のファイル名を設定してください。

③ recv ﾌｧｲﾙ名

recv の PU プログラム用のファイル名を設定してください。

④ ﾃﾞﾌｫﾙﾄ通信ﾁｬﾝﾈﾙ

main プログラム内にﾃﾞﾌｫﾙﾄﾁｬﾝﾈﾙ番号にしたチャンネルを選択してください。

確認ボタンで選択されているチャンネルの情報を見ることができます。

⑤ 通信ﾁｬﾝﾈﾙの選択

ﾃﾞﾌｫﾙﾄのﾁｬﾝﾈﾙを実行時に変更したい場合には、“画面問合せをする”を選択してください。

“画面選択をしない“を選択するとﾃﾞﾌｫﾙﾄﾁｬﾝﾈﾙが通信ﾁｬﾝﾈﾙとして使用するようになります。

⑥ 通信ｻｲﾄﾞ

通信ｻｲﾄﾞ（ﾎｽﾄか装置側か、どちらの立場でｼﾐｭﾚｰｼｮﾝを行うか）を選択します。指定されたｻｲﾄﾞの通信用ﾌﾟﾛｸﾞﾗﾑが自動生成されます。□両方の通信を選択すると、実行画面で、ｻｲﾄﾞの切り替えで、両方の立場のｼﾐｭﾚｰｼｮﾝを行うことができます。

⑦ ﾚﾎﾟｰﾄﾌｧｲﾙ名

実行開始時に、画面で問い合わせるかどうかを選択します。問い合わせるを選択すると、実行時にﾚﾎﾟｰﾄﾌｧｲﾙ名の入力画面が表示されますので、そこで指定することができます。

⑧ ﾎﾟｰﾄﾌｧｲﾙの作成選択

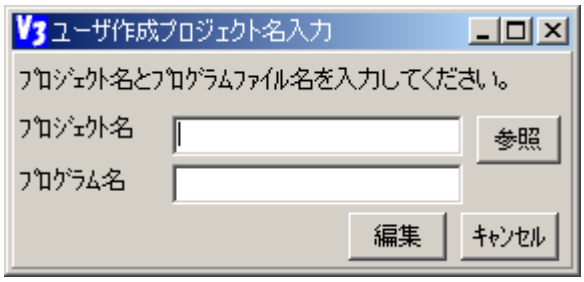
前に行ったｼﾐｭﾚｰｼｮﾝのﾚﾎﾟｰﾄﾌｧｲﾙが残っている場合、ﾚﾎﾟｰﾄﾌｧｲﾙに上書きするか、それとも後ろに追加書き込みするか、または、ﾚﾎﾟｰﾄﾌｧｲﾙを作らないかを指定するための選択指定です。

“画面で問合せする”を選択した場合、実行時にどうするかを選択するための画面を表示するためのプログラムが組み込まれます。

必要な項目の設定が終わったあと、編集かまたは即時実行ボタンをクリックします。

即時実行ボタンの場合は、ただちに実行パネルが表示され、実行を開始します。
==>実行パネルの説明参照。

【ユーザ作成の場合】



① ﾌﾟﾛｼﾞｪｸﾄ名
ﾃﾞﾌｧﾙﾄで、ﾒｯｾｰｼﾞﾌｧｲﾙ名が表示されますが、付けたい名前を入力してください。

② ﾌﾟﾛｸﾞﾗﾑ名
main のPUプログラム保存ファイル名を設定してください。

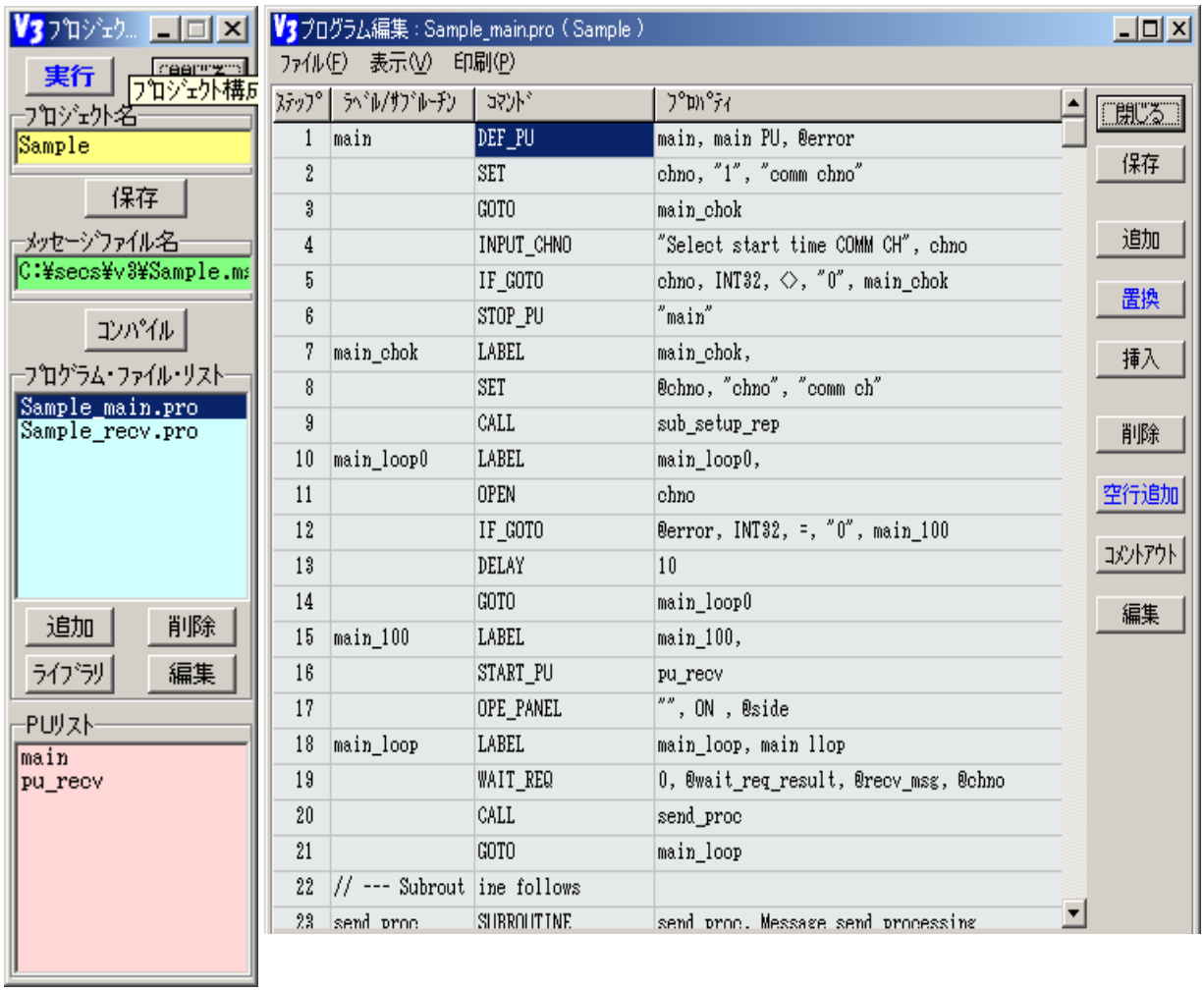
③ 編集をｸﾘｯｸすると、ﾌﾟﾛｼﾞｪｸﾄ編集画面が表示されます。

（5）プロジェクト編集画面

自動生成の場合は、main と recv の２つのﾌﾟﾛｸﾞﾗﾑが生成された上で、ﾌﾟﾛｼﾞｪｸﾄ編集画面に進みます。

A. 自動生成の場合の画面例

sample.msg でﾃﾞﾌｫﾙﾄ画面の設定そのままで生成した場合の例です。



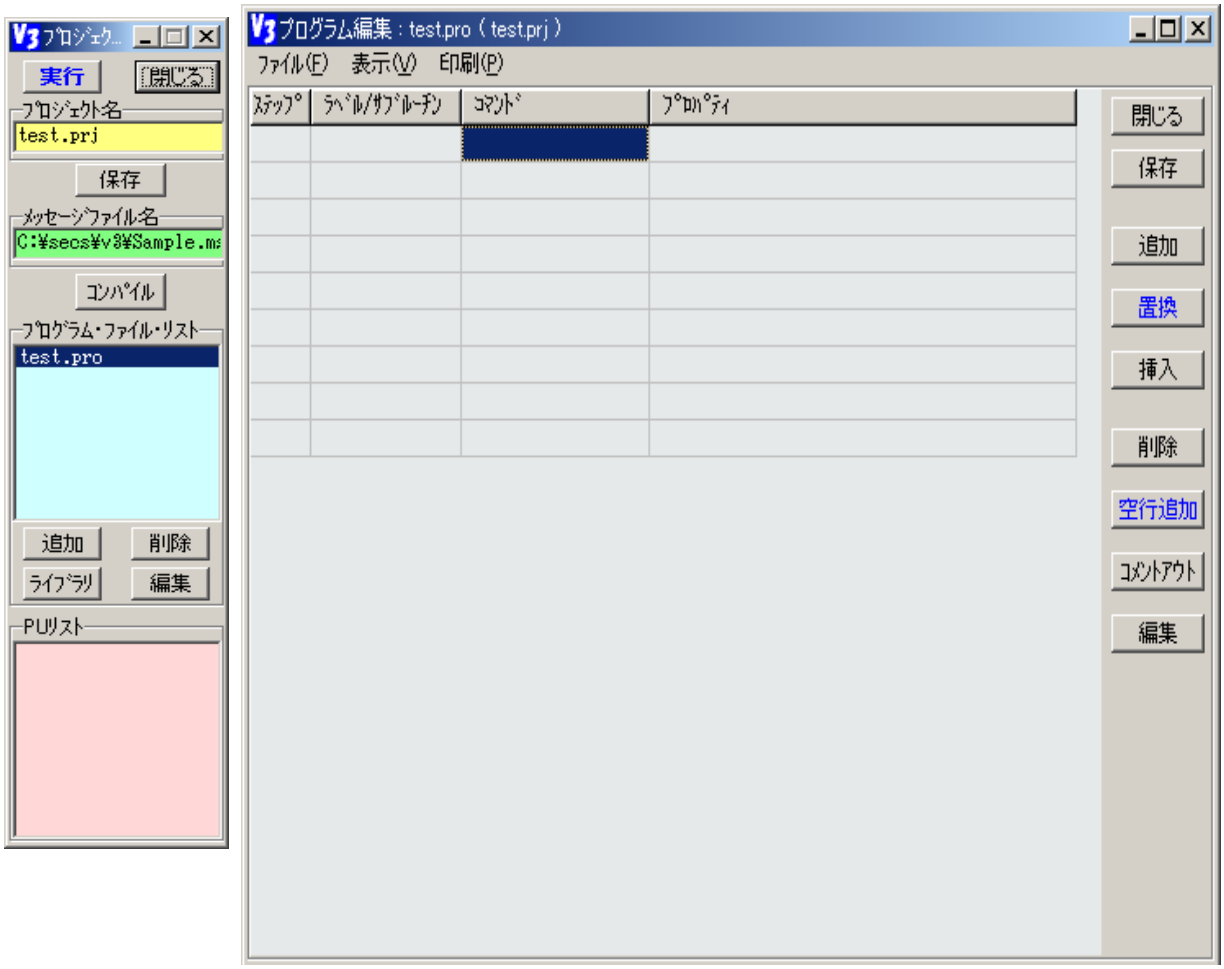
ステップ	ラベル/サブルーチン	コマンド	プロパティ
1	main	DEF_PU	main, main PU, @error
2		SET	chno, "1", "comm chno"
3		GOTO	main_chok
4		INPUT_CHNO	"Select start time COMM CH", chno
5		IF_GOTO	chno, INT32, <>, "0", main_chok
6		STOP_PU	"main"
7	main_chok	LABEL	main_chok,
8		SET	@chno, "chno", "comm ch"
9		CALL	sub_setup_rep
10	main_loop0	LABEL	main_loop0,
11		OPEN	chno
12		IF_GOTO	@error, INT32, =, "0", main_100
13		DELAY	10
14		GOTO	main_loop0
15	main_100	LABEL	main_100,
16		START_PU	pu_recv
17		OPE_PANEL	"", ON , @side
18	main_loop	LABEL	main_loop, main llop
19		WAIT_REQ	0, @wait_req_result, @recv_msg, @chno
20		CALL	send_proc
21		GOTO	main_loop
22	// --- Subrout	ine follows	
23	send_proc	SUBROUTINE	send_proc, Message send processing

このままで保存する場合は、プロジェクト構成画面の保存ボタンをクリックして保存してください。

ユーザ生成の場合は、プログラム編集画面にはコマンドはありません。ユーザは、ここで、ＰＵプログラムの作成またはライブラリ（ＰＵまたはサブルーチン）の作成ができます。

Ｂ．　ユーザ生成の場合の画面例

たとえば、ﾌﾟﾛｼﾞｪｸﾄ名、ﾌﾟﾛｸﾞﾗﾑ名ともに test とした場合の例です。



以下、プロジェクトの編集、プログラムの編集に進みます。

３．２　プロジェクトの編集

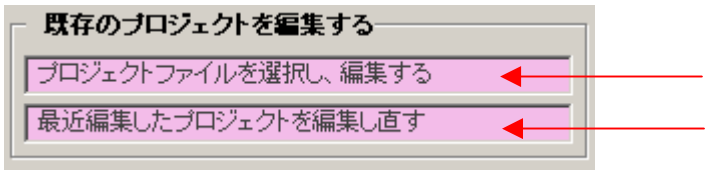
プロジェクトの編集について説明します。

プログラムの編集については、プログラムの編集を、また、編集後の実行についてはプロジェクトの実行操作を参照してください。

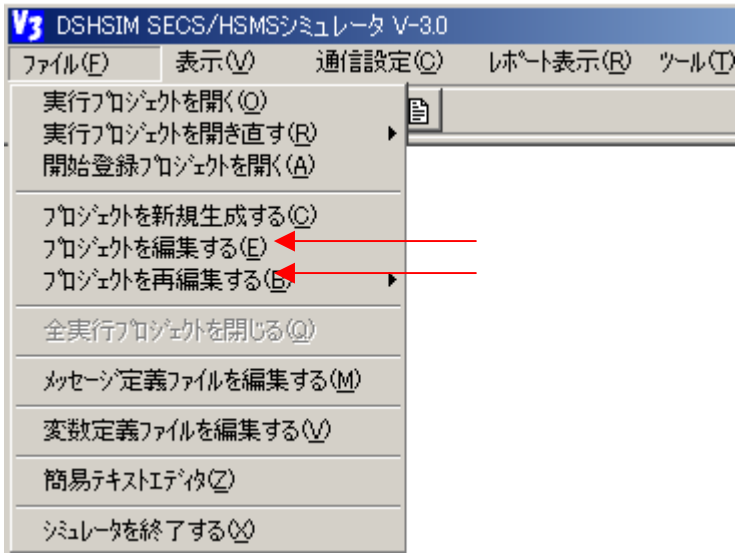
（1）プロジェクトの編集を始める

簡単メニューからの開始とメイン画面のファイルメニューのどちらから開始します。

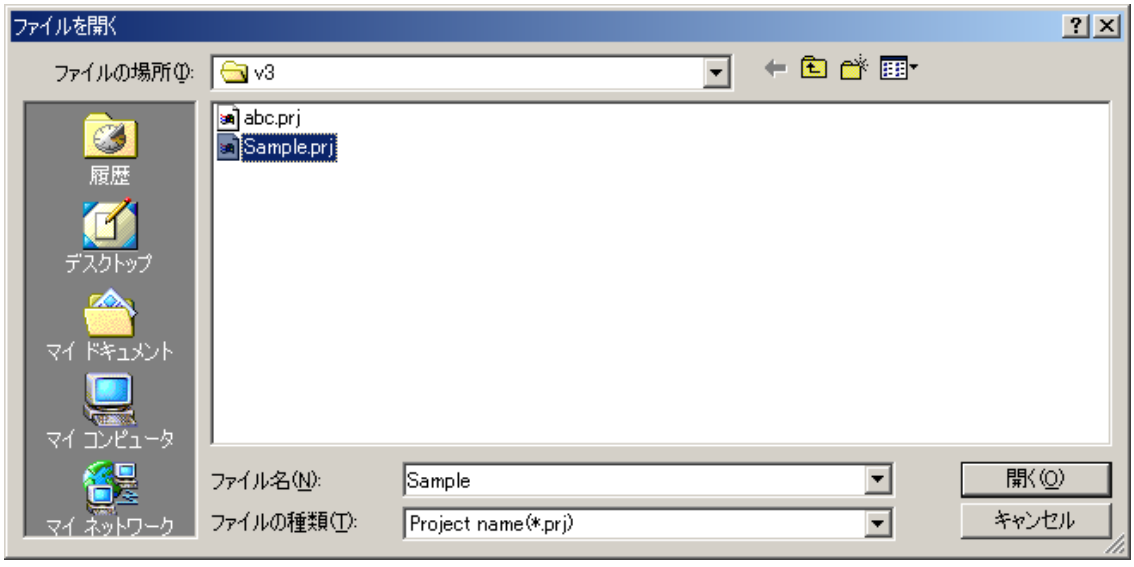
①　簡単メニューからの場合



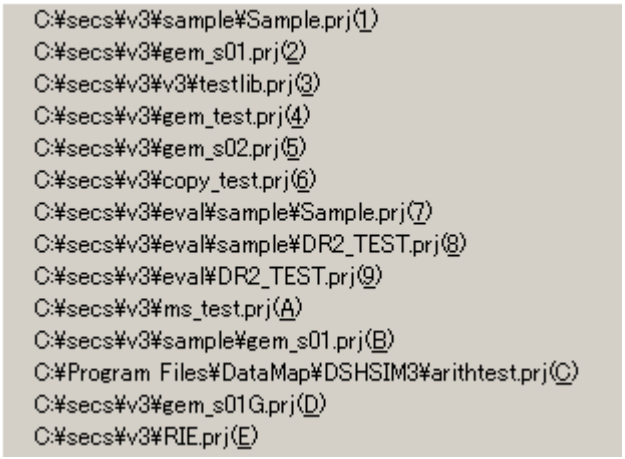
②　メイン画面のﾌｧｲﾙ(F)メニューのﾌﾟﾛｼﾞｪｸﾄを編集する(E)または、ﾌﾟﾛｼﾞｪｸﾄを再編集する(B)タブのクリック、または、ツールボタンの  のクリックから始めます。



プロジェクト編集の場合、ファイルを開くダイアログ画面でプロジェクトを選択します。



ここで、たとえば、Sample を選択し、開くボタンをクリックします。
また、**プロジェクトを再編集(B)**の場合は、以前開いたプロジェクト名がサブメニューとして一覧表示されますので、そこで、編集したいプロジェクト名を選択します。



以前編集のため開いたことがあるﾌﾟﾛｼﾞｪｸﾄ名ﾘｽﾄです。

開かれたプロジェクトの構成は次のようにプロジェクト構成画面に表示されます。
（新規に作成したい場合は、プロジェクトの新規作成を参照ください。）

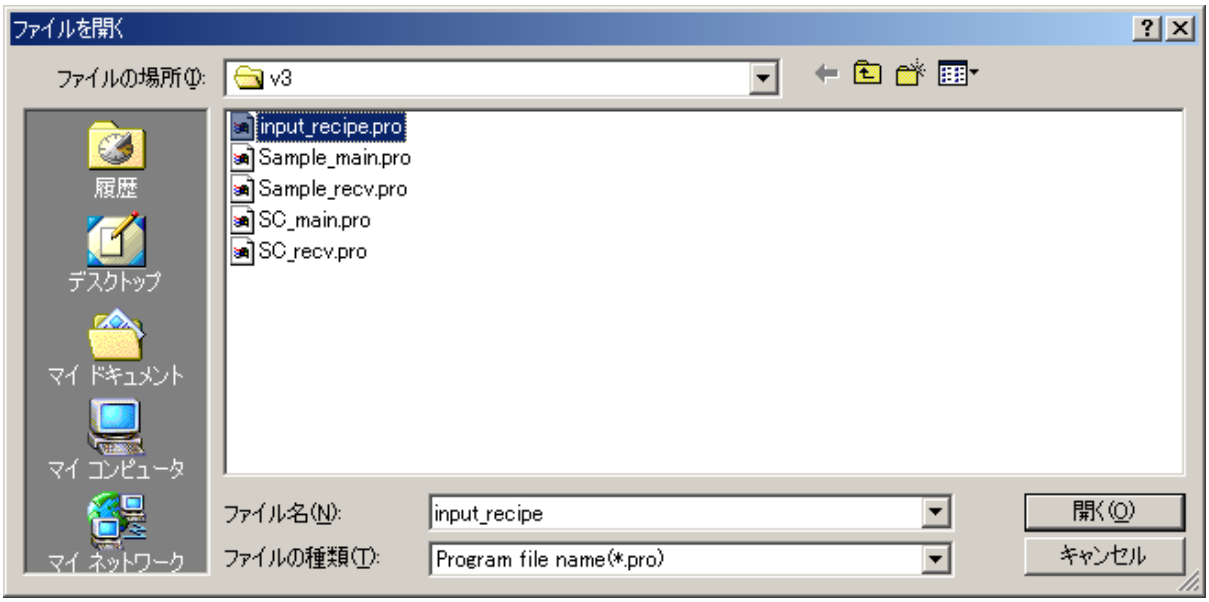
ﾌﾟﾛｼﾞｪｸﾄ名　：ﾌﾟﾛｼﾞｪｸﾄの新規作成操作で指定された名前です。
ﾒｯｾｰｼﾞﾌｧｲﾙ名　：SECS 通信ﾒｯｾｰｼﾞﾌｧｲﾙ名で、新規作成操作で指定された名前です。
ﾌﾟﾛｸﾞﾗﾑ・ﾌｧｲﾙ・ﾘｽﾄ：構成するﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ名です。追加、削除、ﾌﾟﾛｸﾞﾗﾑの編集ができます。
PU ﾘｽﾄ　：ﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ内に含まれる PU 名の一覧です。

プロジェクトのＰＵ構成編集は、**ボタン**の操作で行う追加、削除、ﾗｲﾌﾞﾗﾘ、編集があります。

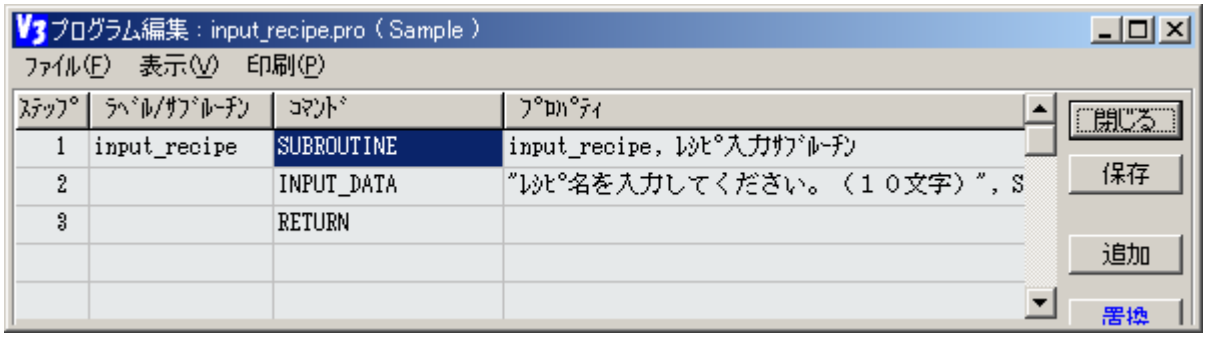
（2）プログラムファイルを追加する操作

たとえば、既に作成されているプログラム、**input_recipe.pro** ファイルを追加したい場合には次のように操作します。

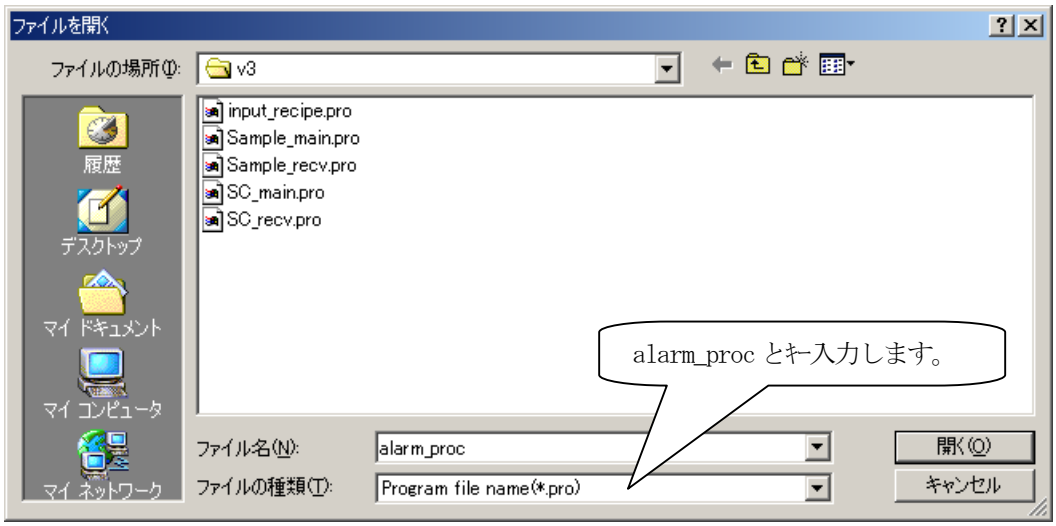
① 追加ボタンをクリックします。**ファイルを開く**ダイアログ画面が表示されますので、ここで、input_recipe.pro を選択し、開くボタンをクリックします。



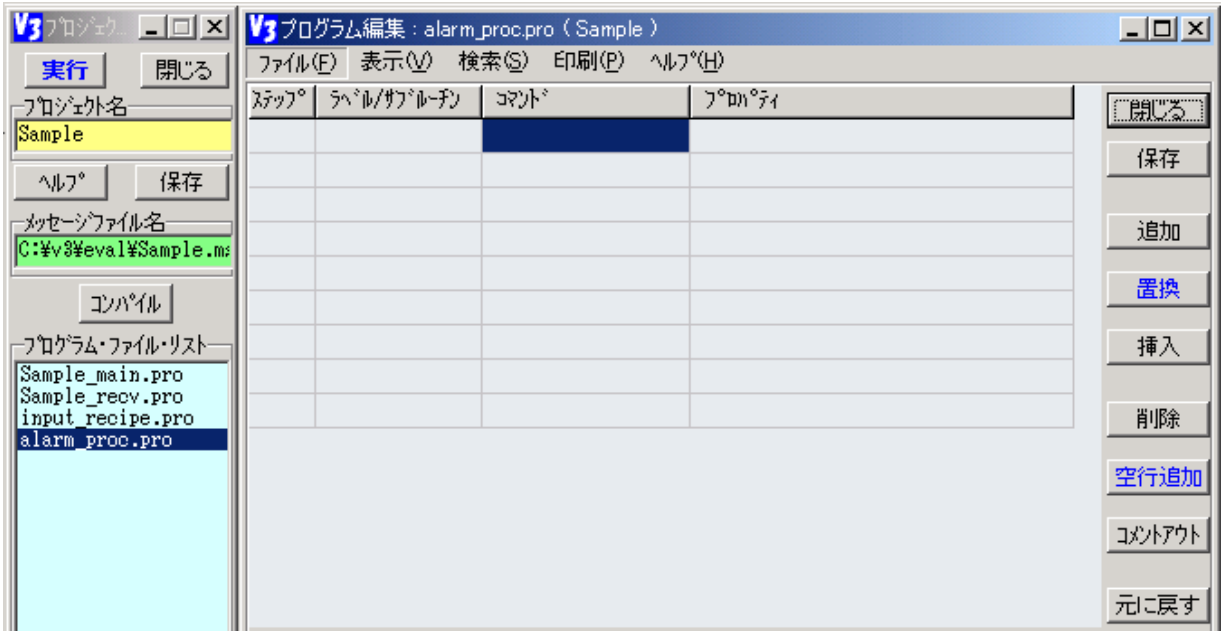
プロジェクト構成画面は下のようになります。追加した input_recipe.pro がﾌﾟﾛｸﾞﾗﾑ・ﾌｧｲﾙ・ﾘｽﾄ欄に追加され、プログラム編集画面が現れます。



新規にプログラムファイルを作成する場合は、**ファイルを開く**ダイアログ画面で、新規に作成したいプログラム名を入力し、開くボタンをクリックします。たとえば、alarm_proc.pro を作成する場合には、alarm_proc と入力し、開きます。



すると、プロジェクト構成画面に alarm_proc.pro が追加され、コマンドが空のプログラムファイル alarm_proc.pro の編集画面が現れます。



（3）プログラムファイルを削除する操作

プロジェクト構成に含まれているプログラムファイルの削除は、次の手順で行います。

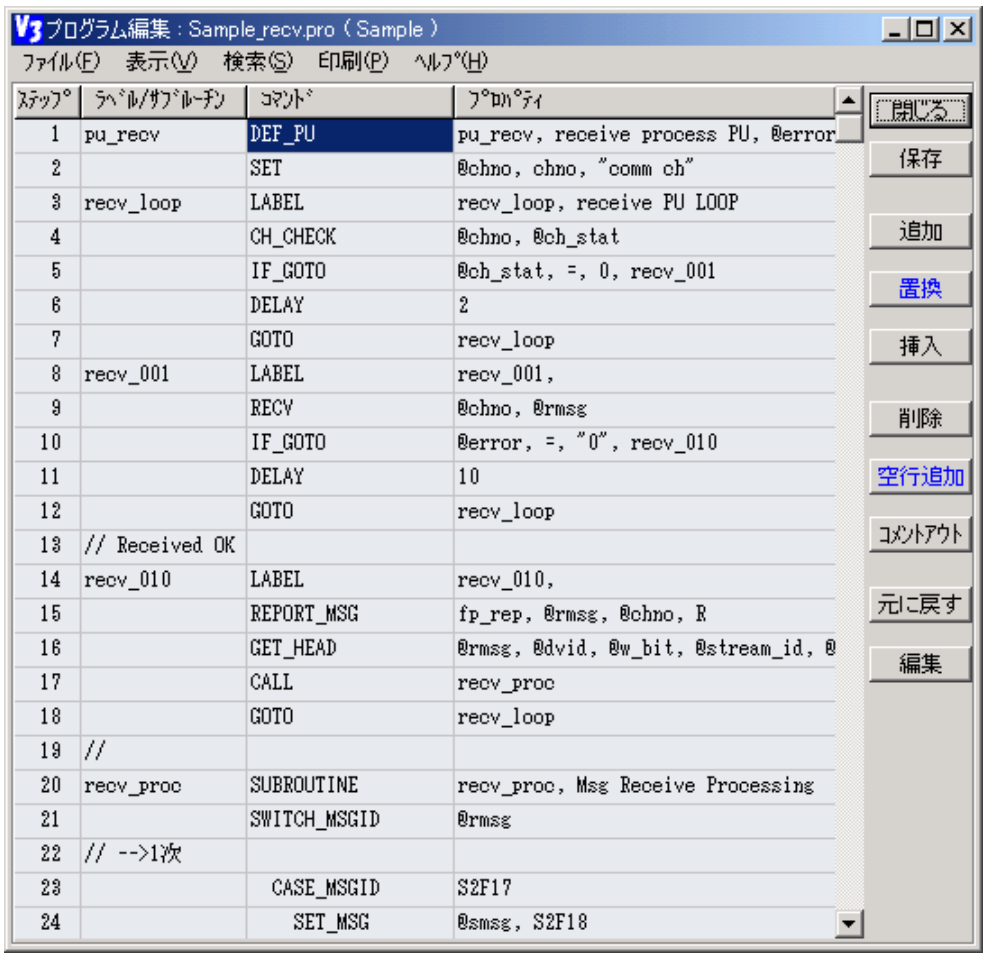
最初に、ﾌﾟﾛｸﾞﾗﾑ・ﾌｧｲﾙ・ﾘｽﾄの中の削除したいプログラムファイル名をクリックして選択します。

たとえば、alarm_proc.pro を削除する場合は、次の画面のように alarm_proc.pro を選択します。

次に、削除ボタンをクリックします。すると、削除確認画面がポップアップされますので、はい(Y)をクリックしてください。
上の alarm_proc.pro が構成から削除されます。

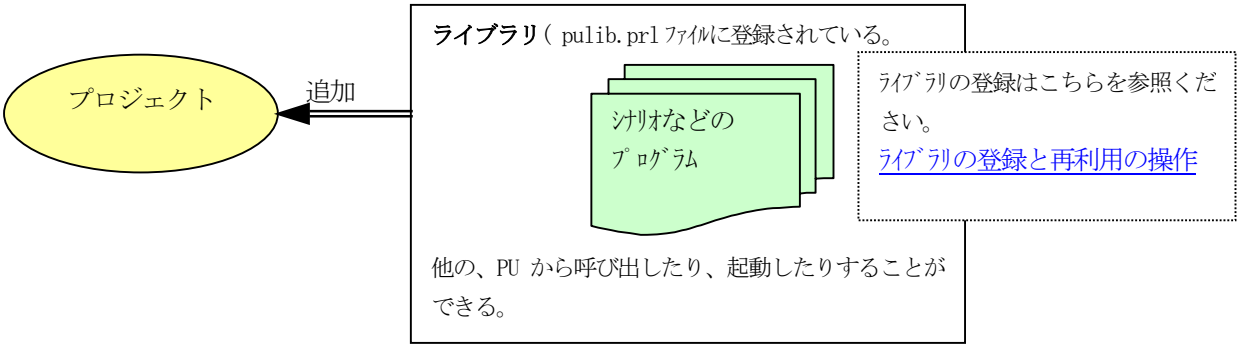
(4) プログラム編集画面を表示する。

まず、編集したいプログラムファイル名をﾌﾟﾛｸﾞﾗﾑ・ﾌｧｲﾙ・ﾘｽﾄ上でクリックし、選択します。(削除操作の時のように)
次に編集ボタンをクリックします。たとえば、recv.pro の編集の場合、次のような画面が表示されます。

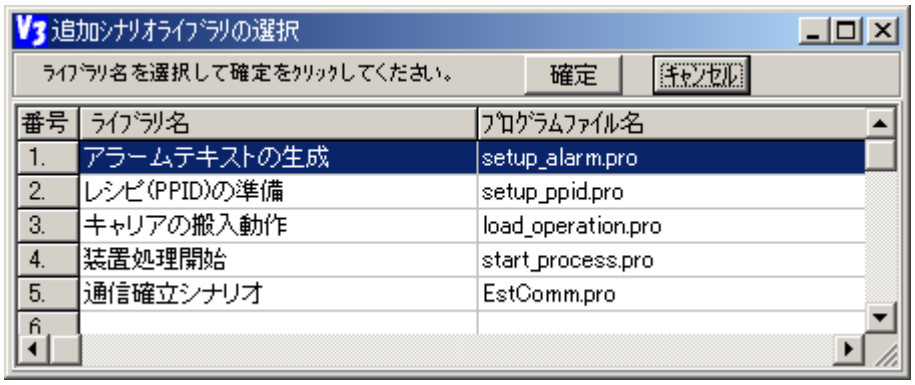


(5) ライブラリプログラムを追加する。

予め、プログラムファイルをライブラリに登録しておき、ライブラリに含まれるプログラムファイルをプロジェクトに追加することができます。



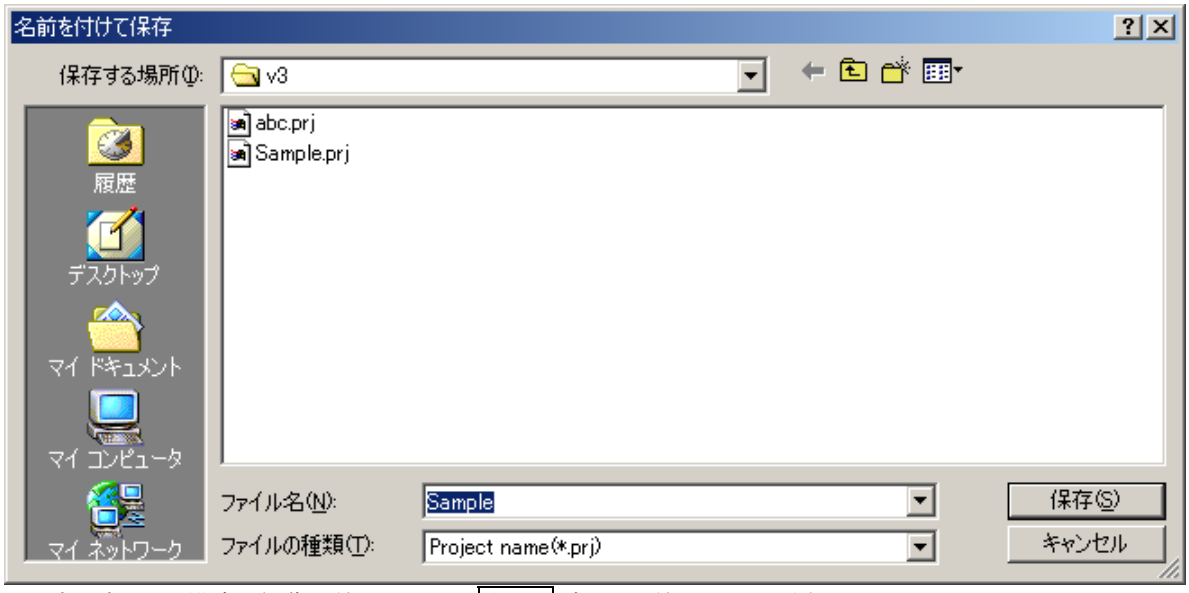
まず、ﾗｲﾌﾞﾗﾘボタンをクリックすると、ライブラリに登録されているプログラムの一覧表の画面が表示されます。



ここで、加えたいライブラリ名を選択し、確定ボタンをクリックします。
たとえば、1 番目のアラームテキストの生成を選択、確定すると、setup_alarm.pro が加えられ、構成画面は次のようになります。

（6）プロジェクトの保存操作

プロジェクトの構成の変更があった場合は、あらためてプロジェクトを保存する必要があります。
保存は、ﾌﾟﾛｼﾞｪｸﾄ保存ボタンのクリックで始めます。
次の名前を付けて保存ダイアログ画面が表示されますので、ここで、プロジェクト名をセットして保存ボタンをクリックして保存します。

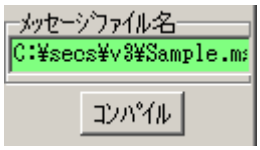


プロジェクト構成の編集が終わったら、閉じるボタンで終了してください。

（7）メッセージファイルを変更した場合の再コンパイル

プロジェクトを構成するメッセージファイルが更新された場合は、通常、シミュレータが自動的に再コンパイルしてくれますが、意識してコンパイルしたい場合は、再コンパイルすることができます。

再コンパイルは、メッセージファイル名の表示の右隣りのｺﾝﾊﾟｲﾙボタンのクリックで行います。



（8）編集中プロジェクトの実行

編集中のプロジェクトを実行したい場合は、実行ボタンをクリックします。

もし、プロジェクトが未保存だったりした場合は、画面の指示に従って保存してから実行することになります。

この実行は、メイン画面メニューからの実行と同じ働きをします。

３．３　プログラムの編集

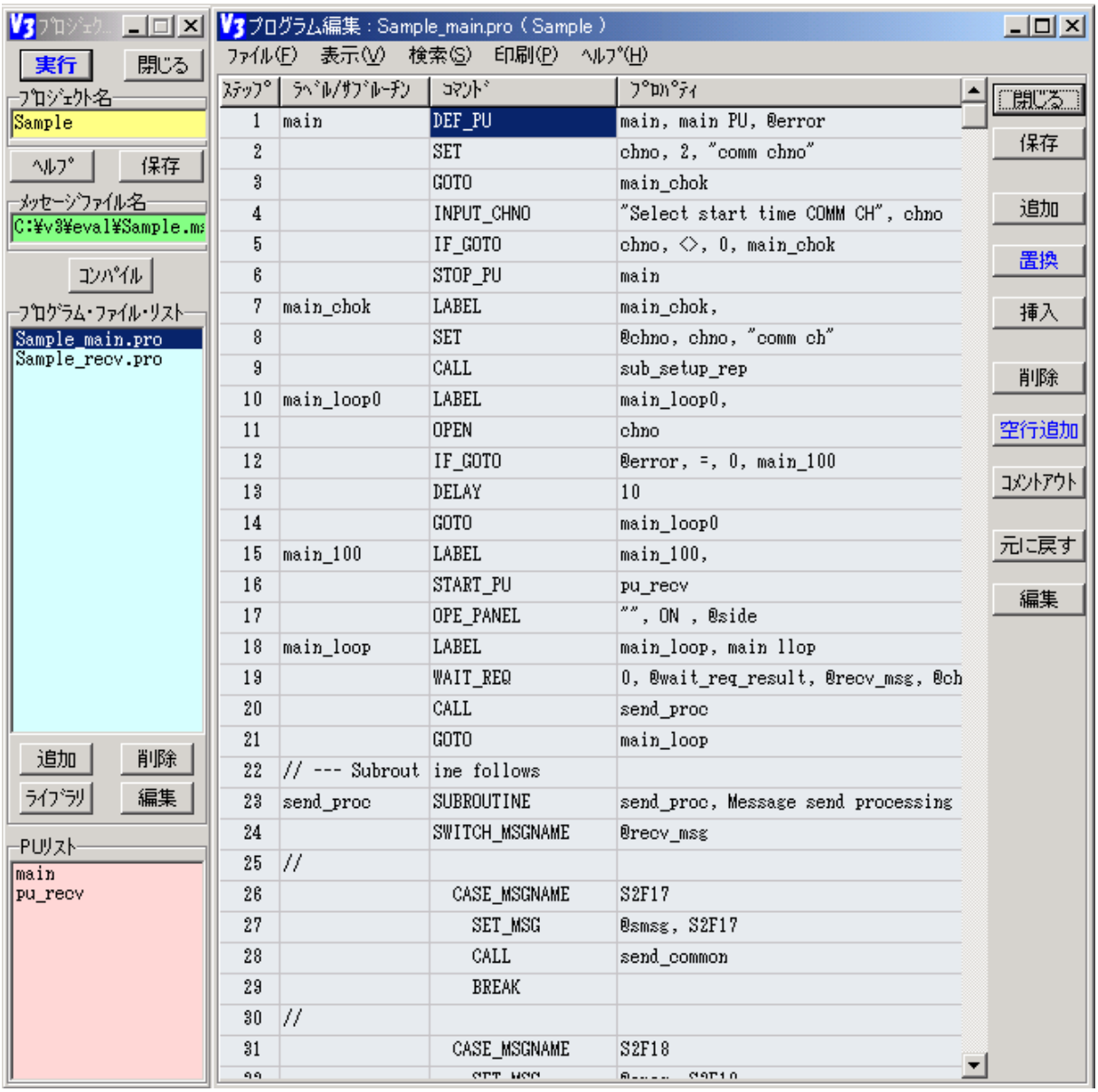
プログラムの編集は、プロジェクト編集画面のサブ画面として、プロジェクトを構成するプログラムの編集操作を行います。

（参照：　プロジェクトの編集）

（1）編集画面の表示

プロジェクト構成画面で、構成するプログラムを選択し、編集ボタンのクリックでプログラム編集画面を表示させます。(注) プロジェクト編集が開かれた際、１番目のプログラムファイルのための編集画面が自動的に表示されます。

たとえば、自動生成された Sample.prj の Sample_main.pro の編集画面は次のようになります。

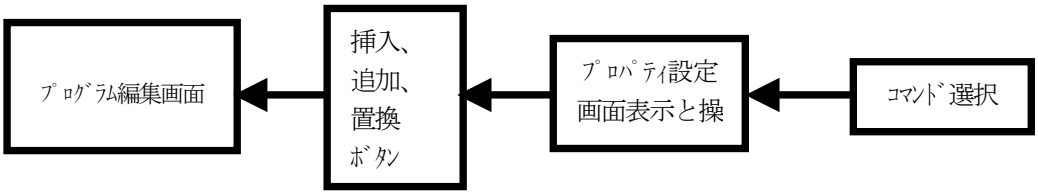


（2）コマンドの編集（挿入、追加、置換）

コマンドの編集には、挿入、追加、置換（上書き）そして削除の４つの種類があります。

挿入、追加、上書きは、コマンドのプロパティ設定画面との連携で行います。

基本的に、次の流れになります。



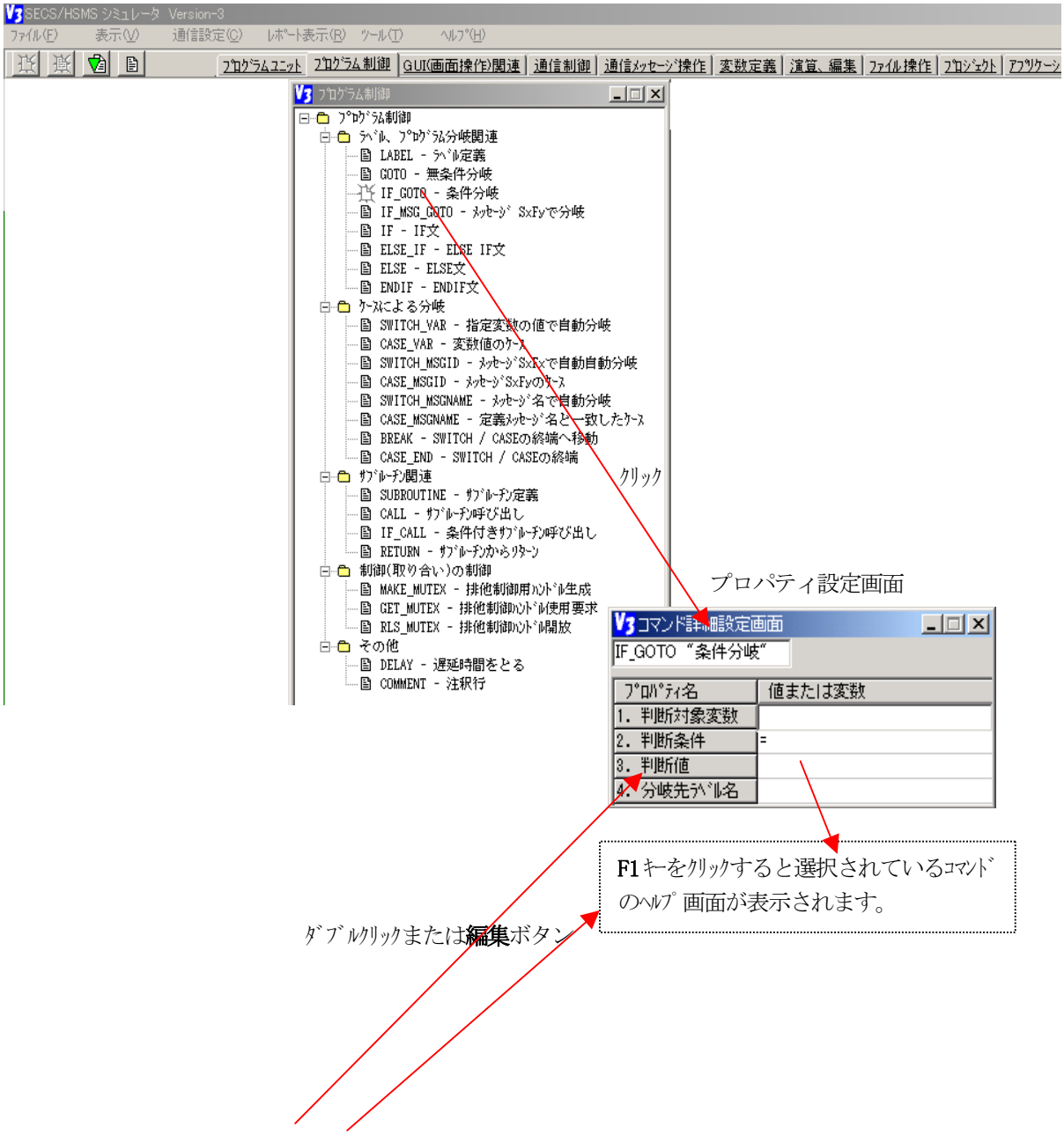
① 編集コマンドの選択とプロパティ設定画面表示

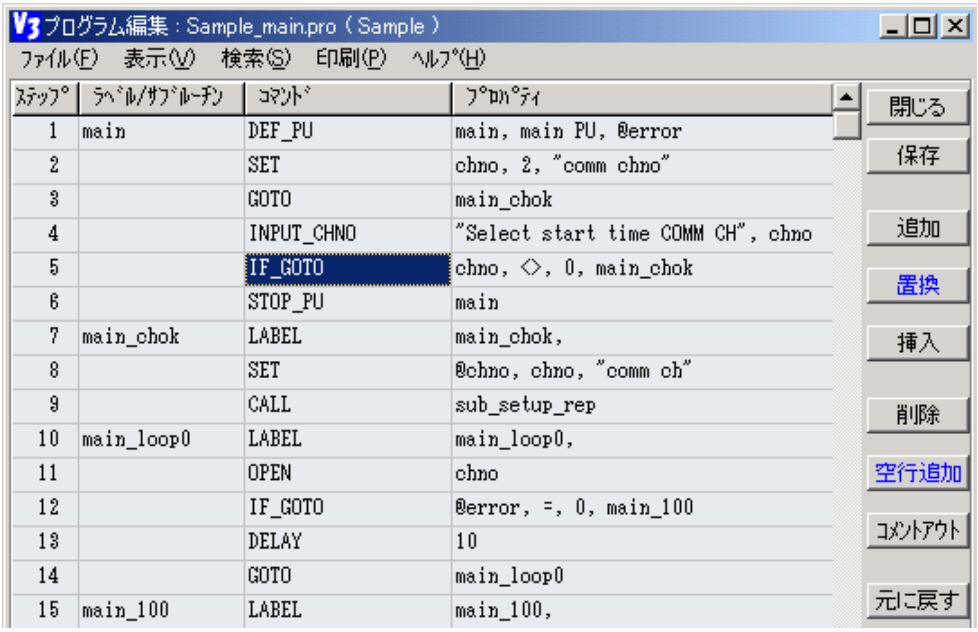
編集対象のコマンドの選択には、２つの方法があります。

・メイン画面のコマンドタブからの選択

・プログラム編集画面上のコマンドの選択

たとえば、 下の図は、IF_GOTO コマンド編集のためのプロパティ設定画面を出す２つの操作を示しています。



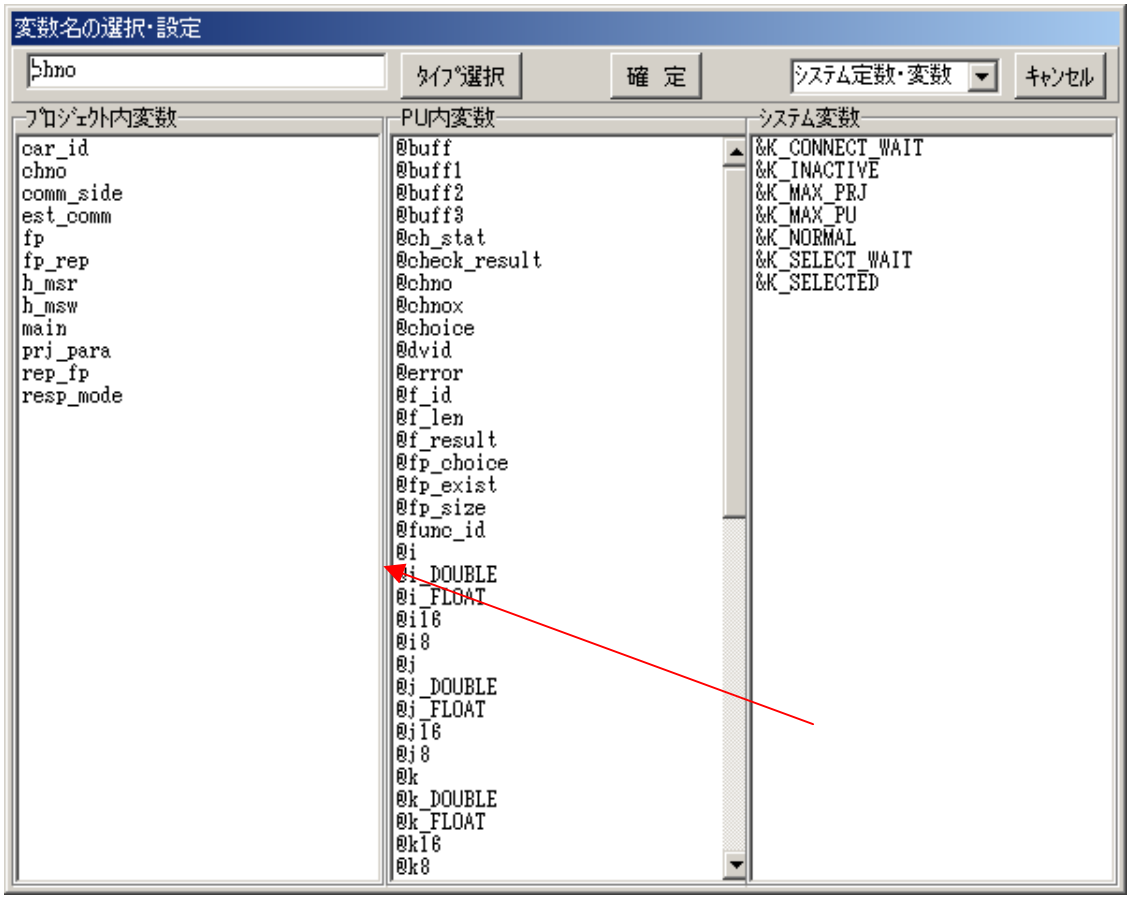


② プロパティ設定画面の操作

たとえば、IF_GOTO コマンドのプロパティ設定の場合、下の画面の 4 つのプロパティを設定します。1 番から順に設定します。プロパティの設定は、そのタイプによってできるだけ値や名前の選択が画面上でできるように設計されています。

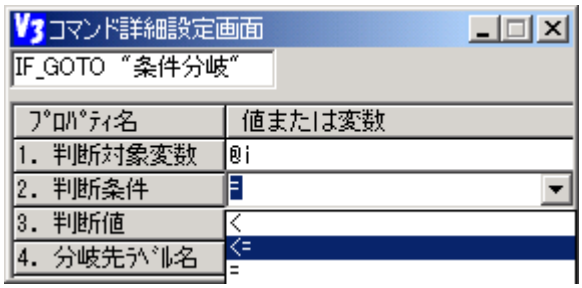
以下、順に、プロパティを設定していきます。

1 の**判断対象変数**の**値または変数名**の欄をダブルクリックすると、変数として登録されて変数名選択画面が表示されます。



ここで、設定したい変数名を選択し、**確定**ボタンをクリックすると、それが、1. の判断変数の値として設定されます。たとえば、@i を選択し、確定すると次のようになります。

2 の判断条件は、クリックするとｺﾝﾎﾞﾎﾞｯｸｽが表示されますので、その中の 1 つを選択します。次のような選択画面になります。



3 の判断値は、直接、値を設定するか、または変数名を設定します。変数名の場合は、1 の変数選択と同じように選択確定します。

4 の分岐先ﾗﾍﾞﾙ名については、プロジェクト内に定義されているラベル名の一覧表から選択することができます。未定義のラベルを指定する場合は、設定したいラベル名を直接キー入力で設定し、あとで、

そのラベル名をラベル定義コマンドで定義するようにします。

ﾗﾍﾞﾙ名選択画面です。
変数選択と同様に選択し、確定ﾎﾞﾀﾝを
ｸﾘｯｸします。

以上の結果、たとえば、プロパティ設定画面は、次のようになります。

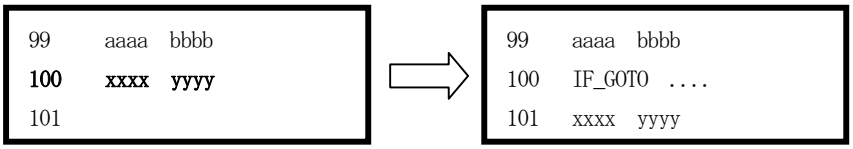
③　ロパティ設定画面のコマンドを挿入、追加、上書き

挿入、追加、上書きボタンのクリックでプログラム編集画面に反映させます。

各ボタンの機能は、現編集位置が、１００ステップ目だと仮定すると、次のようになります。

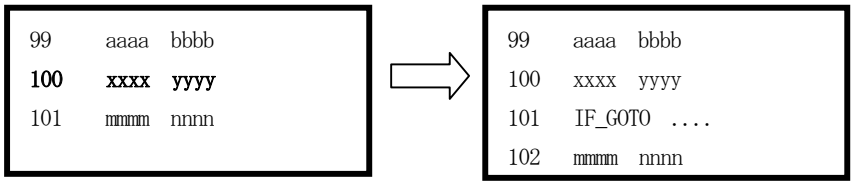
挿入　：　現在開かれている位置の前に挿入されます。

新しい IF_GOTO コマンドがステップ 100 に入り、元の 100 ステップ以降のコマンドが 101 番目以降にシフトダウンされます。

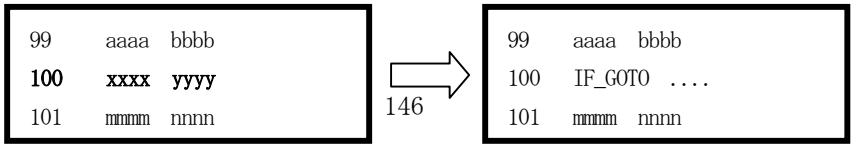


追加　：　現在開かれている位置の次の行に追加挿入されます。

元の 101 ステップ以降のコマンドが後方にシフトダウンされます。

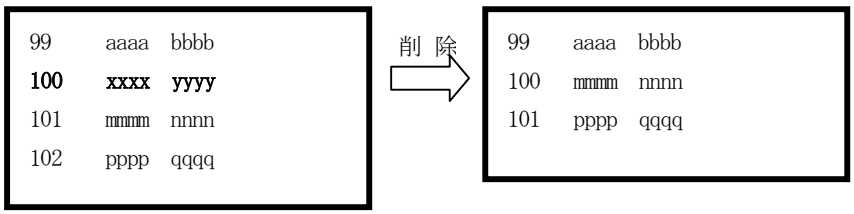


上書き：　現在開かれている位置に上書きされます。元のコマンドは消失します。



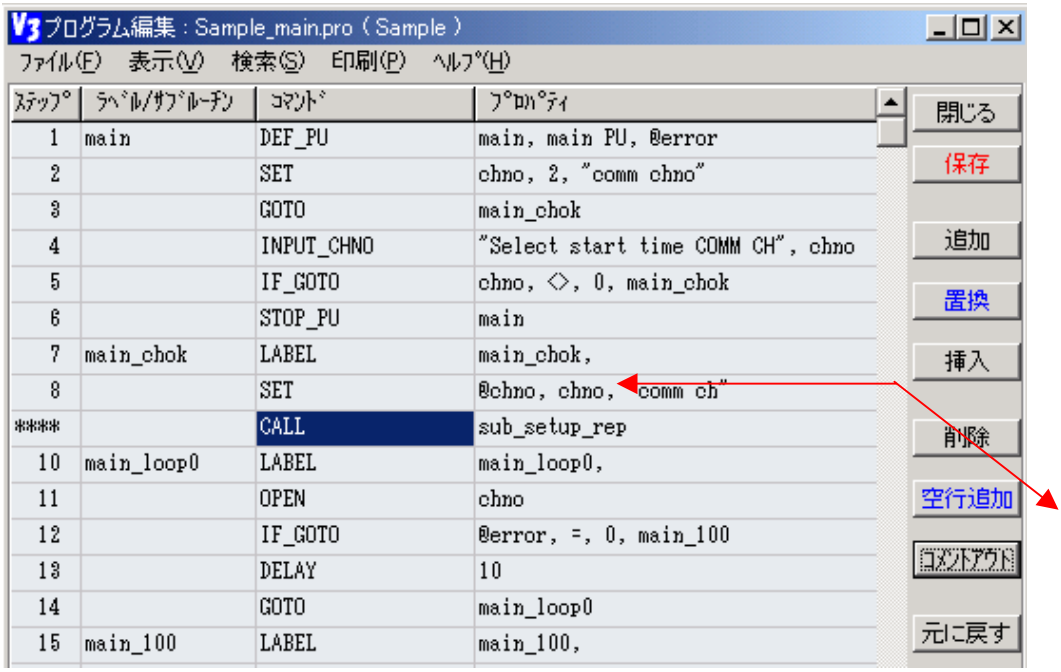
（3）コマンドの削除

コマンドの削除は、プログラム編集画面で削除したいコマンドのステップ位置を選択し、削除ボタンをクリックすることによって行います。



（4）コマンドのコメントアウト

コマンドのコメントアウトは、コメントアウトしたいコマンドを選択し、ｺﾒﾝﾄｱｳﾄボタンをクリックすることによって行います。コメントアウトするとそのコマンドのステップ番号が **** 表示になります。
コメントアウトされたコマンドは、実行時に実行されないで、スキップします。
下の画面では、CALL sub_setup_rep コマンドがコメントアウトされた状態を表しています。実行時、このCALL コマンドは実行されません。



（5）空行追加

プログラムを見やすくするため、わざと空の行を入れたい場合がありますが、そのために使用します。
空行を追加したいステップのコマンドを選択し、**空行追加**ボタンをクリックします。

（6）編集を元に戻す

元に戻すボタンを使って追加、削除など直前に行った編集（変更）を取り消して元に戻すことができます。
このボタンを連続的にクリックすると、過去にさかのぼって記憶されている範囲内で順番に編集を取り消すことができます。
元に戻すことができる編集操作は次のとおりです。

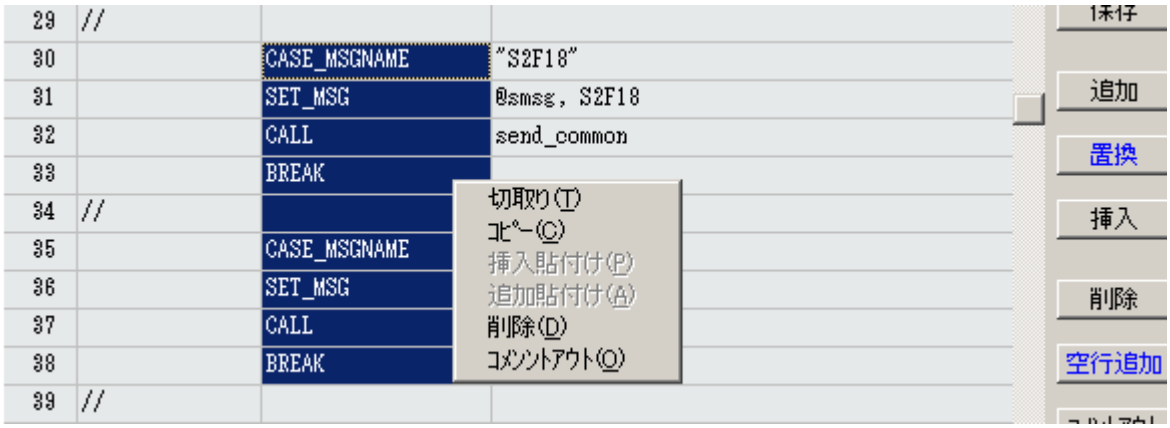
① **追加、置換、挿入、削除、空行追加**ボタンによる正常動作
② （8）で述べるブロック編集の中の、**切取り、挿入貼付け、追加貼付け、削除**の４つの機能
（7）の保存が行われると、過去の編集動作の記憶は消去されます。

（7）プログラムの保存

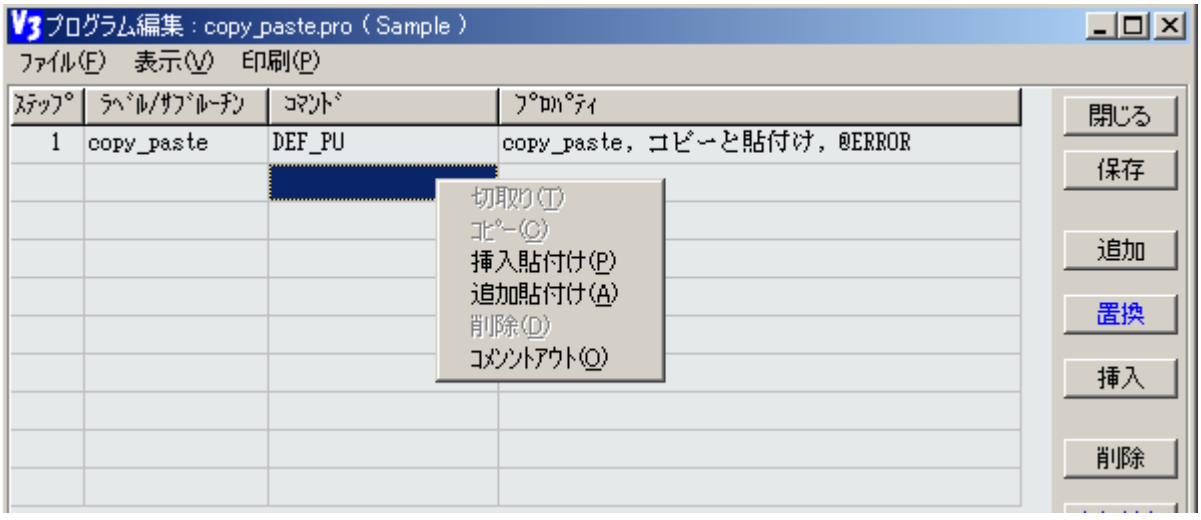
編集作業が済んだら、**上書き保存**ボタンのクリックで変更後のプログラムを保存します。
プログラムの変更操作が行われると**保存**ボタンの色が赤になります。

（8）ステップのブロックコピー、ブロック削除

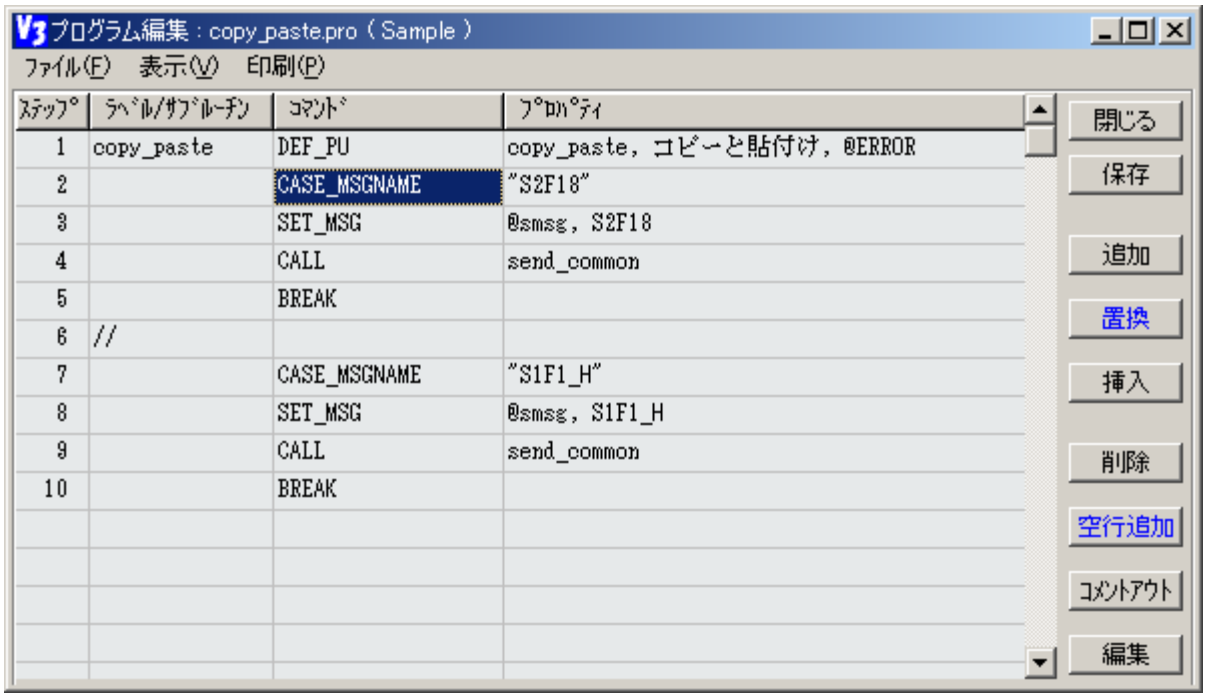
プログラム内の連続したコマンド群をコピー（COPY）し、自プログラム内または他のプログラムに貼付け（PASTE）することができます。
操作は次のとおりです。
① コピーしたいコマンド群の先頭をクリックし選択します。
② 次に SHIFT キーを押したまま、コピーしたいコマンド群の最後をクリックします。
③ これで、範囲が選択されたことになります。
④ そして、右クリックし、編集ポップアップ画面を出し、**コピー(C)**をクリックします。
これで、コマンド群がシミュレータ内のクリックボードに格納されます。
（ここで、切り取り(T)をクリックすると、コマンド群は切り取られ、クリップボードに格納される）



⑤ 次に、コピー先のステップを選択します。
⑥ そして、右クリックで、編集画面をポップアップし、挿入貼付け(P)または、追加貼付け(A)をクリックします。
挿入貼付けと追加貼付けの違いは、（2）-□の編集で説明した、挿入と追加の場合と同じです。



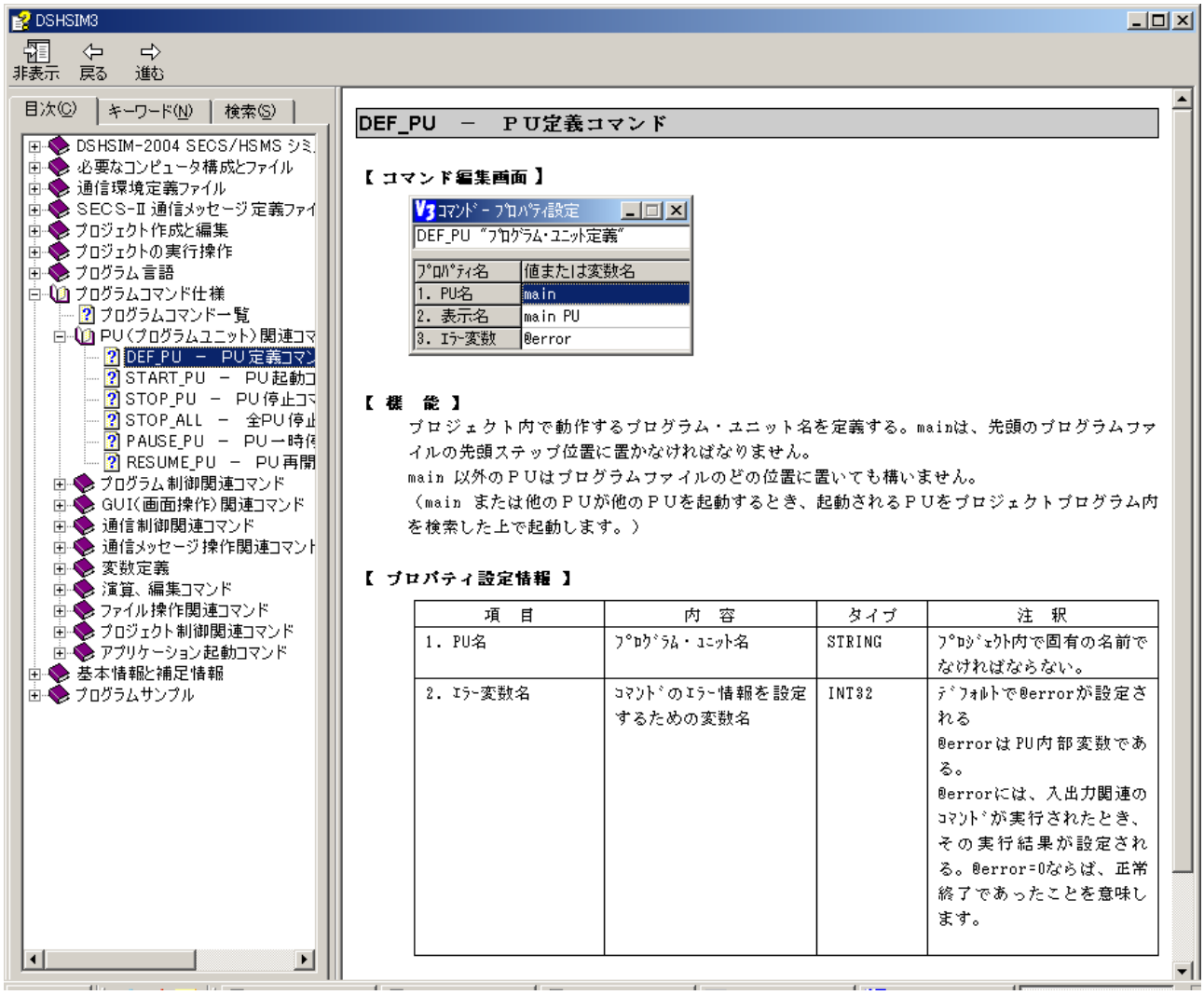
次の画面は、追加貼付けした場合の結果画面です。



⑦　ブロックの削除は、□の編集ポップアップ画面で**削除(D)**をクリックで行います。
削除された場合はクリップボードには、なにも格納されません。

（9）コマンドヘルプの表示

プロパティ設定画面、プログラム編集画面で**Ｆ１**（ﾌｧﾝｸｼｮﾝｷｰ1）を押すと、表示または選択されているコマンドのヘルプ画面を開き、コマンドの詳細を参照することができます。



(10) 変数などの選択設定画面

コマンドのプロパティ値に設定する変数などを選択する画面です。プロパティ値入力欄のダブルクリックでプロパティタイプに対応する選択画面が表示され、画面の一覧表から目的の変数などを選択するために使用します。

① PU 選択画面

関連コマンド
- PU_START
- PU_STOP
- PU_PAUSE
- PU_RESUME

② サブルーチン選択画面

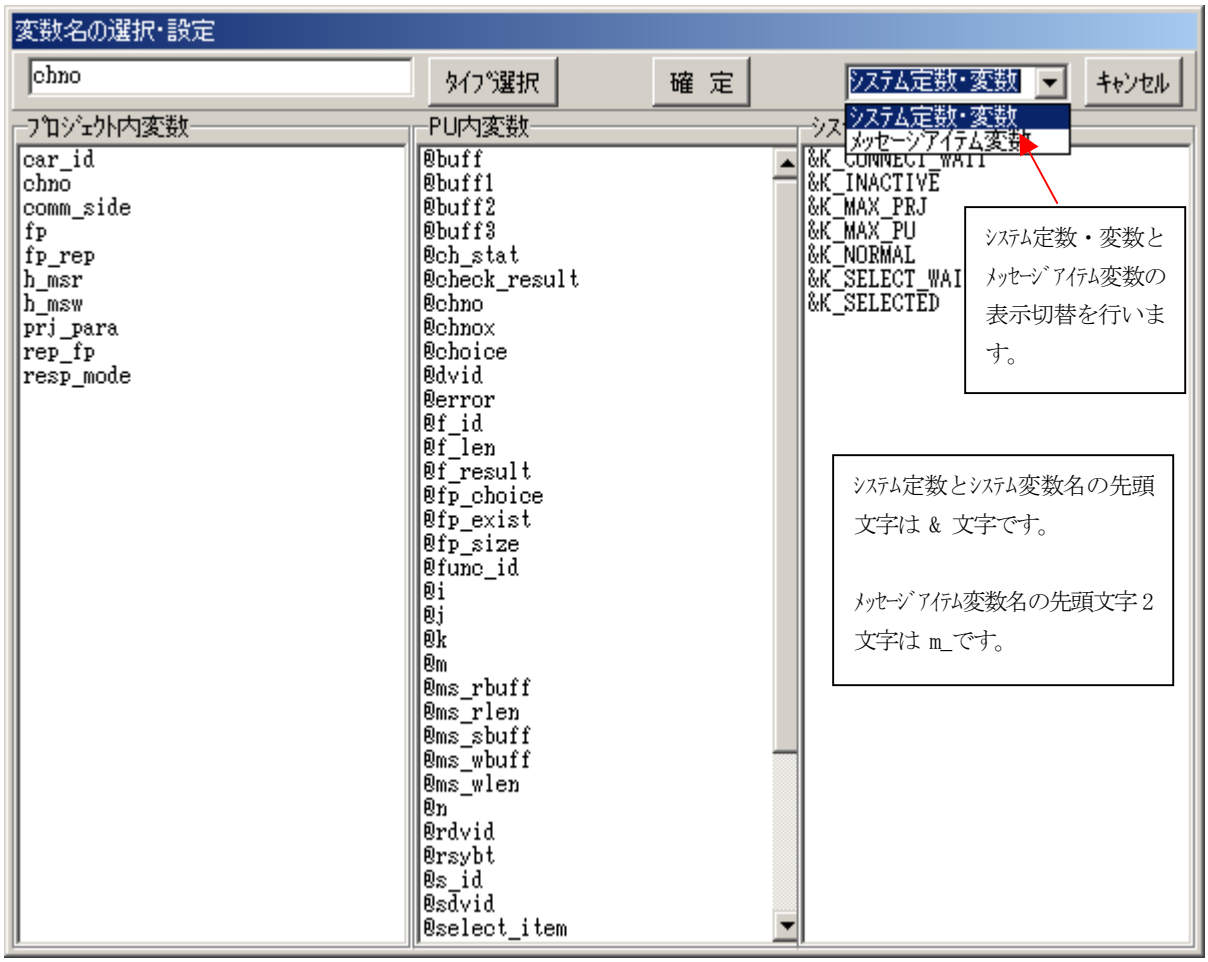
関連コマンド
CALL
IF_CALL

③ ラベル選択画面

関連コマンド
GOTO
IF_GOTO
IF_MSG_GOTO

④ データ変数選択画面

関連コマンドとして、IF_GOTO、IF_CALL、SET、SWITCH_VAR その他のコマンドがあります。

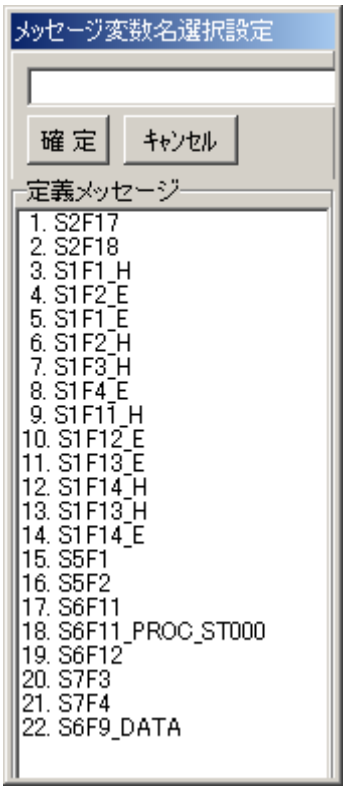


⑤　メッセージ変数選択画面

⑥　SECS メッセージ ID 選択画面

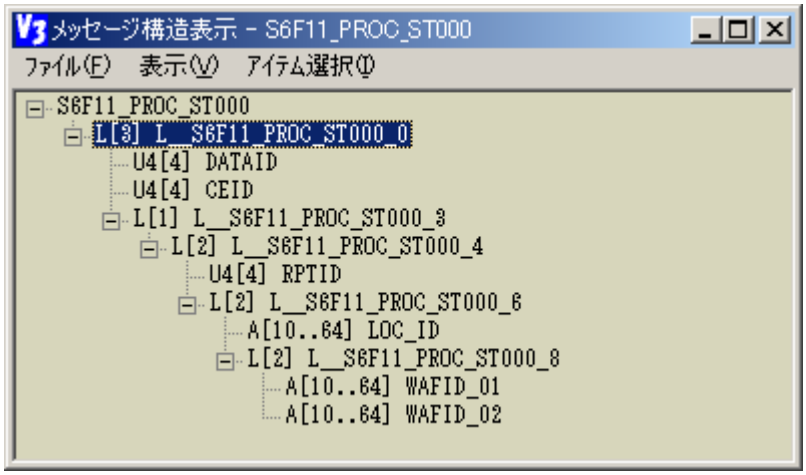
関連コマンド
IF_MSG_GOTO
CASE_MSGID

⑦ SECS 定義メッセージ名選択画面



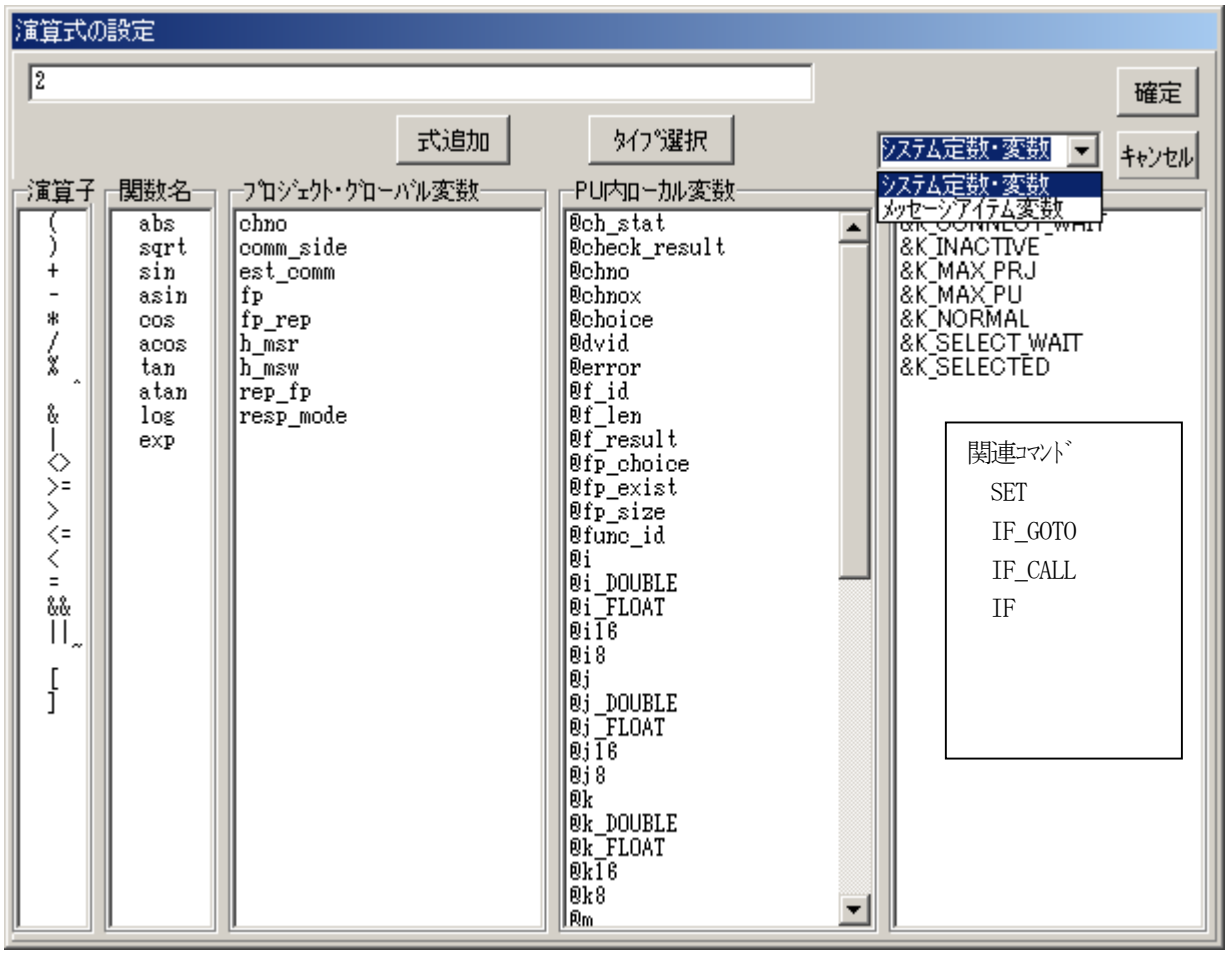
関連コマンド
CASE_MSGNAME
PUT_ITEM_S
PUT_ITEM_P
GET_ITEM_S
GET_ITEM_P

⑧ メッセージアイテム名選択画面



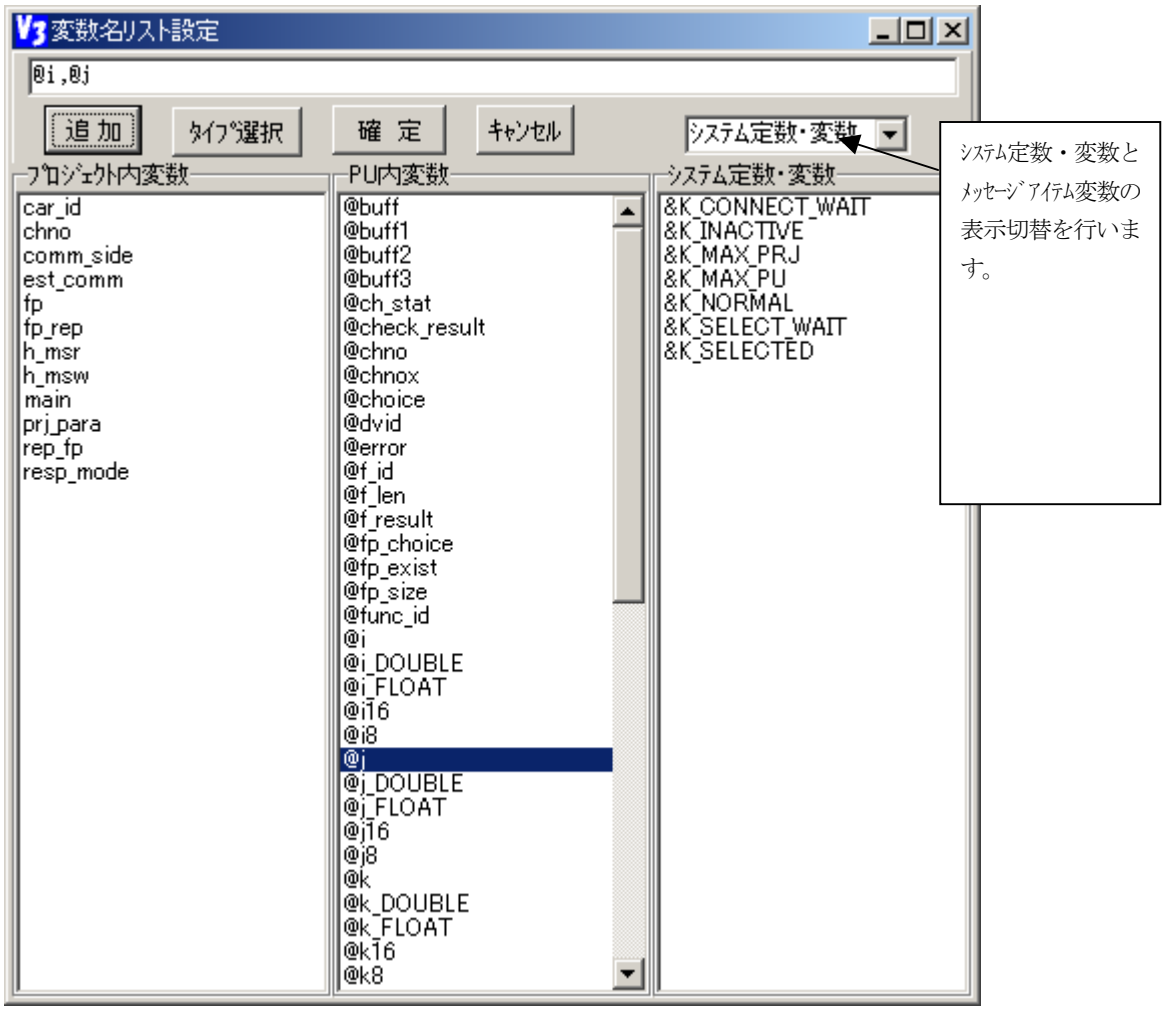
関連コマンド
- PUT_ITEM_S
- PUT_ITEM_P
- GET_ITEM_S
- GET_ITEM_P
- DESIG_RECV

⑨　演算式選択設定画面



⑧　変数名リスト

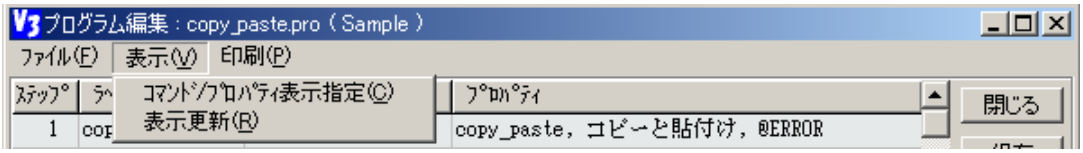
関連コマンドは、FORMAT、OUT_LOG、FPRINTF があります。



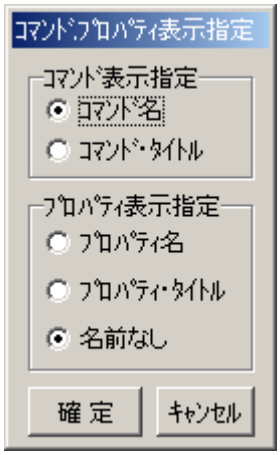
(11) プログラムコマンド、プロパティの表示切り替え操作

プログラム編集画面のプログラムコマンドとプロパティの表示内容を切り替えることができます。

① 操作は、表示(V)メニューの中のｺﾏﾝﾄﾞ/ﾌﾟﾛﾊﾟﾃｨ表示指定タブをクリックします。

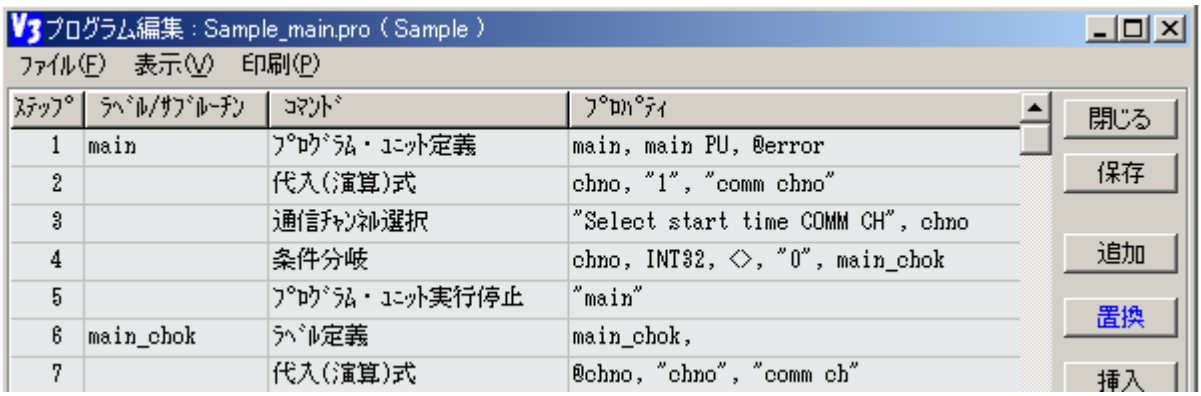


次の画面が表示されます。

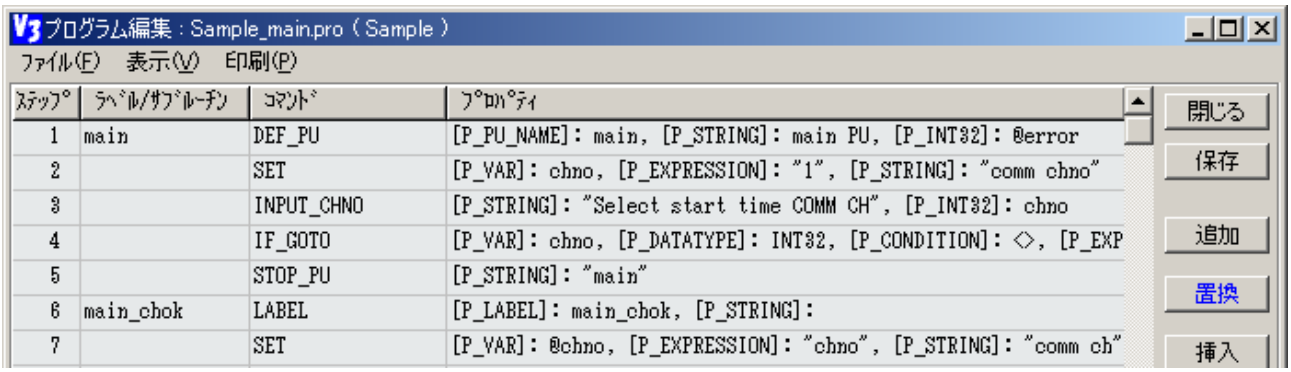


② コマンド表示指定については、ｺﾏﾝﾄﾞ名とｺﾏﾝﾄﾞ・ﾀｲﾄﾙの２つの表示モードがあります。

・ｺﾏﾝﾄﾞ名の場合、通常の表示モードになります。
・ｺﾏﾝﾄﾞ・ﾀｲﾄﾙの場合には、次のようなコマンドのタイトル（CAPTION）が表示されます。



③　ﾌﾟﾛﾊﾟﾃｨ表示指定にはﾌﾟﾛﾊﾟﾃｨ名、ﾌﾟﾛﾊﾟﾃｨ・ﾀｲﾄﾙ、名前なしの３つのモードがあります。
・ﾌﾟﾛﾊﾟﾃｨ名の場合、次のようにﾌﾟﾛﾊﾟﾃｨ名の表示付になります。

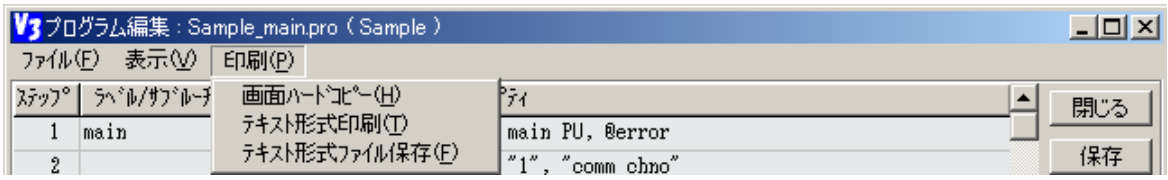


・ﾌﾟﾛﾊﾟﾃｨ・ﾀｲﾄﾙの場合、プロパティのタイトル(CAPTION)付表示になります。

・名前なしの場合は、通常のﾌﾟﾛﾊﾟﾃｨ名やﾀｲﾄﾙなし表示になります。

(12) プログラムの表示情報の印刷とファイル保存

プログラム編集画面に表示されているプログラムの内容をプリンタに印字することができます。また、ファイルに保存することができます。（ここで保存されたファイルは.TXT ファイルであり、ロードできません。
操作は、印刷メニューを使って行います。



① 画面ﾊｰﾄﾞｺﾋﾟｰは、プログラム編集画面をプリンタにそのまま印刷します。

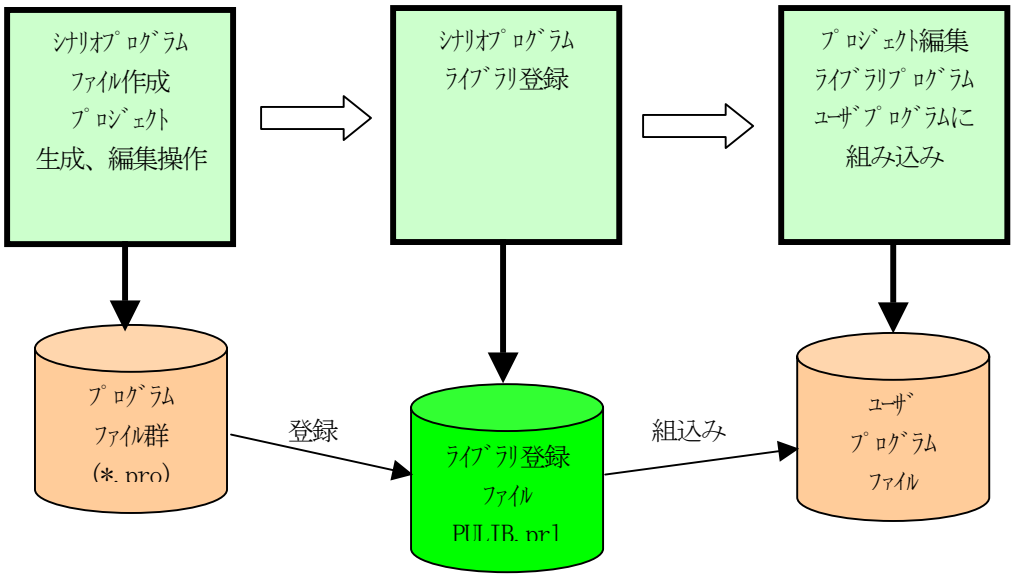
② ﾃｷｽﾄ印刷は、プログラムの全内容をテキスト形式で印刷します。

③ ﾃｷｽﾄ形式ﾌｧｲﾙ保存は、プログラムの全内容をテキスト形式(.TXT)でファイルに保存します。

３．４　ライブラリの登録と再利用の操作

たとえば、ＧＥＭ仕様で使用する通信シナリオプログラムを作成しておき、後で、プロジェクトの中にそれらを組み込み使用することができます。

操作順はつぎのようになります。



① ﾗｲﾌﾞﾗﾘに登録したいﾌﾟﾛｸﾞﾗﾑを仮のﾌﾟﾛｼﾞｪｸﾄで構いませんが、ﾌﾟﾛｼﾞｪｸﾄの生成編集操作の中で作成します。
ﾌﾟﾛｸﾞﾗﾑはPUﾌﾟﾛｸﾞﾗﾑ、ｻﾌﾞﾙｰﾁﾝのどちらでも構いません。

② 次に、ﾗｲﾌﾞﾗﾘ登録をﾒｲﾝ画面のﾂｰﾙﾒﾆｭｰのｼﾅﾘｵﾗｲﾌﾞﾗﾘ登録タブの操作で登録します。

③ 別のﾌﾟﾛｼﾞｪｸﾄを作成する際に、ﾗｲﾌﾞﾗﾘに登録されているｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑを組み込みます。

（1）シナリオプログラムの作成と編集

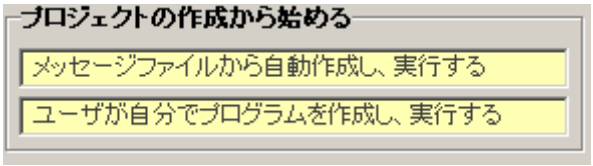
シナリオプログラムの作成と編集は、通常のプロジェクト、プログラムの生成、編集操作で行います。
各操作の詳細については、以下を参照してください。

プロジェクトの新規作成
プロジェクトの編集
プログラムの編集

たとえば、通信接続の例で仮に、EstComm.pro を作成します。使用するメッセージファイルは gem_s01.MSG を使い、通常のプロジェクトを自動作成の操作の流れの中で、作成することにします。

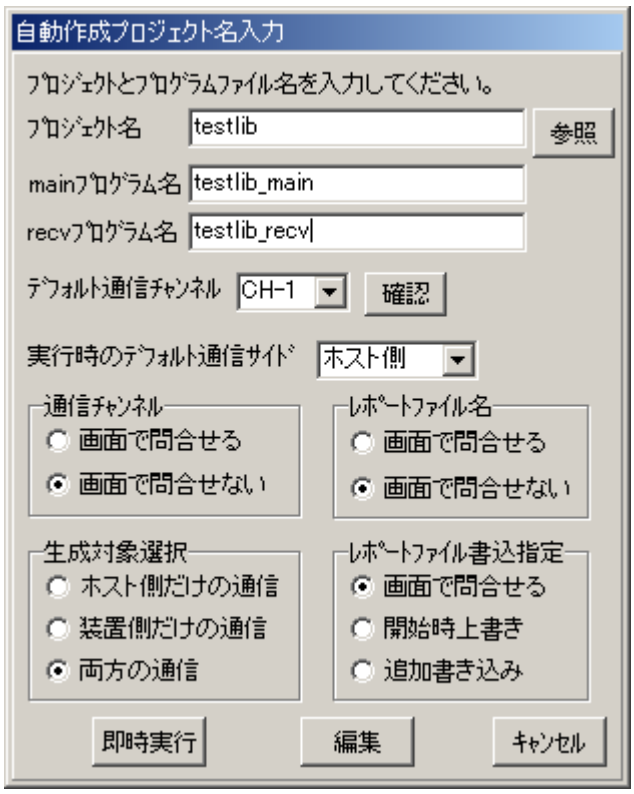
① プロジェクト生成操作を行う。

a. 簡単メニューから

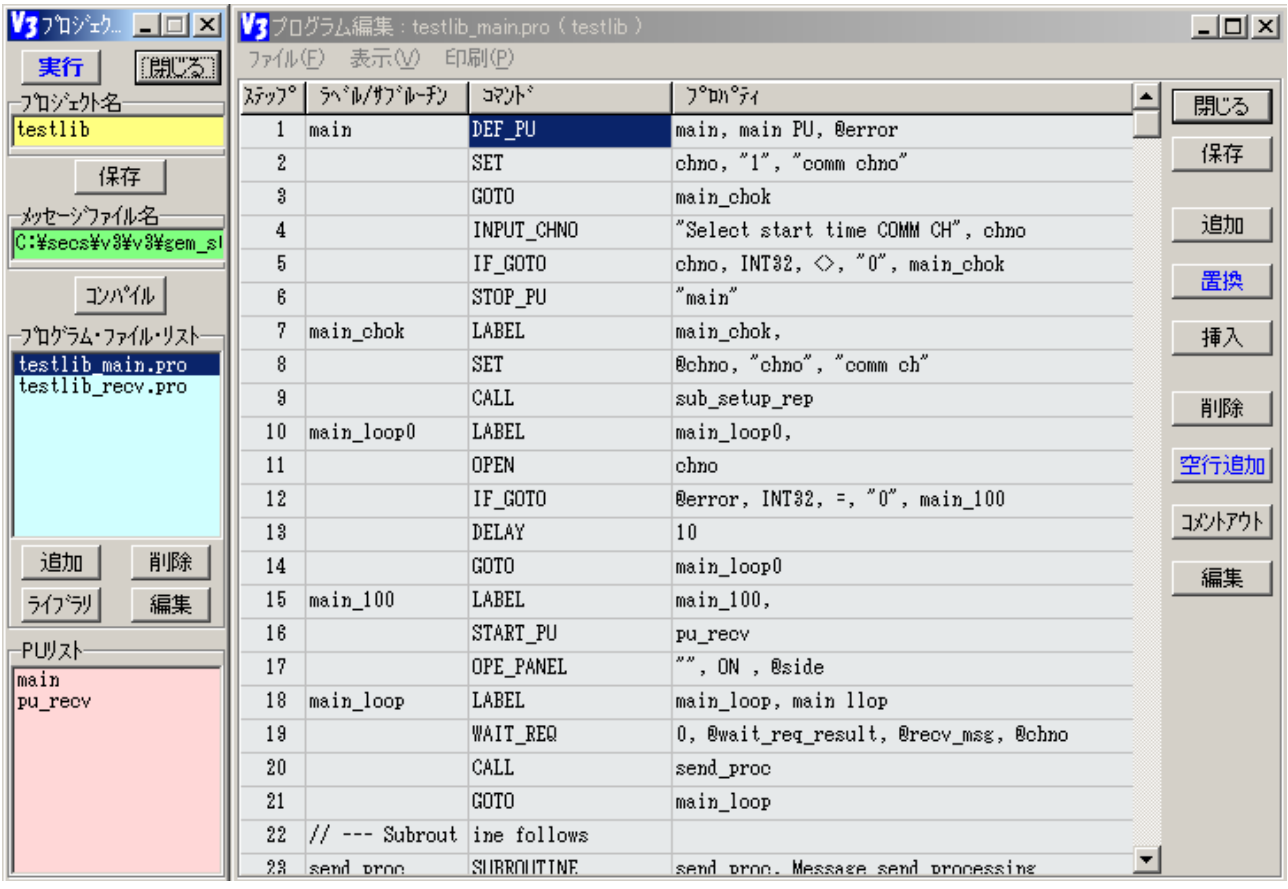


b. メイン画面のﾌｧｲﾙ(F)ﾒﾆｭｰのﾌﾟﾛｼﾞｪｸﾄを新規作成する(C)タブのクリックから始めます。
ﾒｯｾｰｼﾞﾌｧｲﾙとして gem_s01.MSG を選択し、ﾌﾟﾛｼﾞｪｸﾄ名とﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ名を次のようにキー入力設定します。

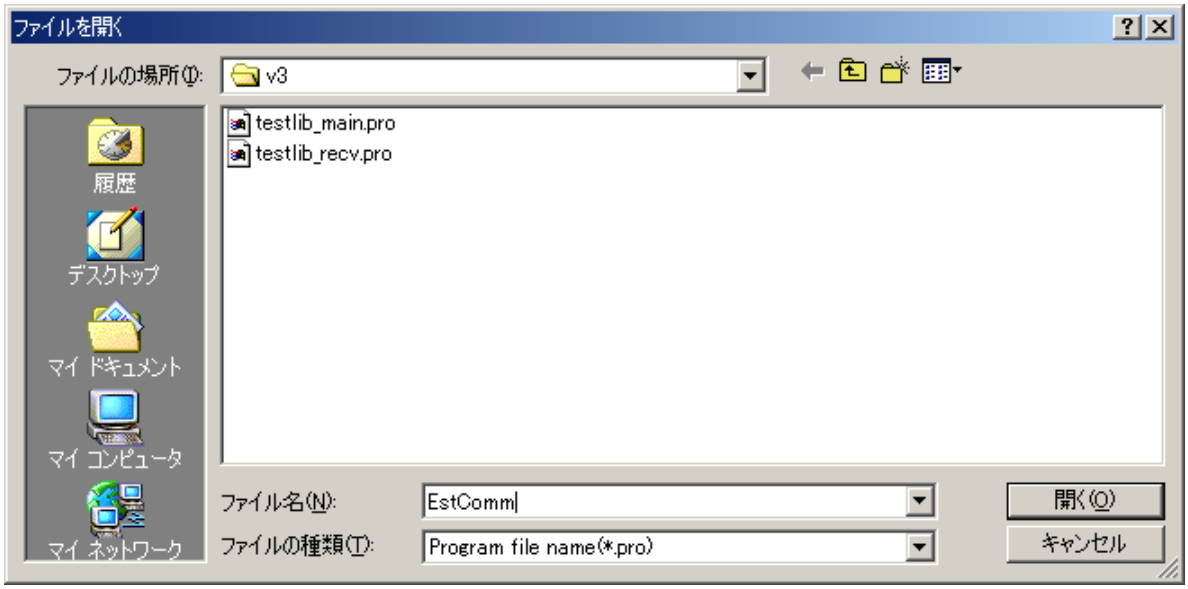
Est_main – (ﾒｲﾝﾌﾟﾛｸﾞﾗﾑ)
Est_recv – (受信処理ﾌﾟﾛｸﾞﾗﾑ)



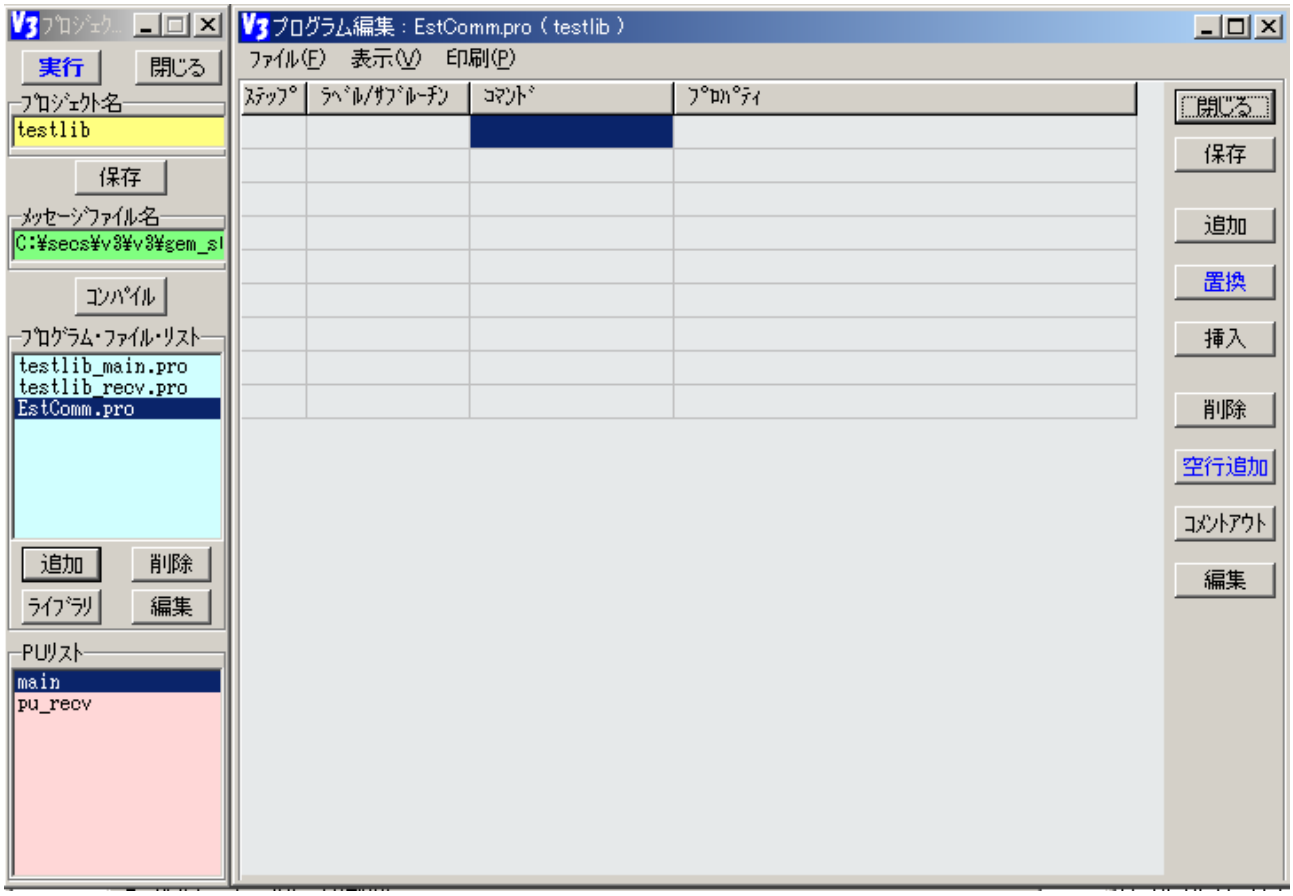
確定の後、編集画面になります。



② 通信確立のためのプログラムファイル EstComm.pro を作成するため、追加ボタンをクリックします。
ファイルを開く画面が出ますので、EstComm と入力し、開く ボタンをクリックします。



③ 編集画面は、次のようになります。



④ これから以下のような簡単な通信シナリオをプログラミングします。

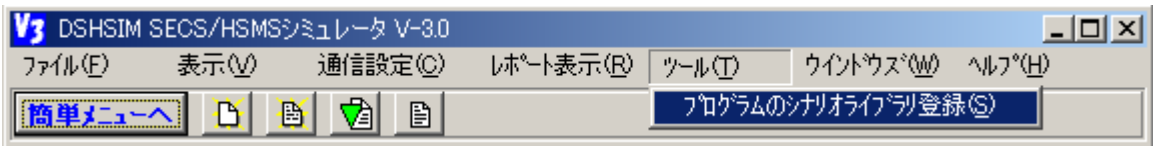
ホスト	通信	装置
S1F13	------>	
	<======	S1F14
S1F17	------>	
	<======	S1F18
	<------	S6F11 CEID=100
S6F12	======>	

プログラムの内容は、シナリオプログラム例を参照してください。
この例では、独立した PU 名 EstComm として動作するようになっています。

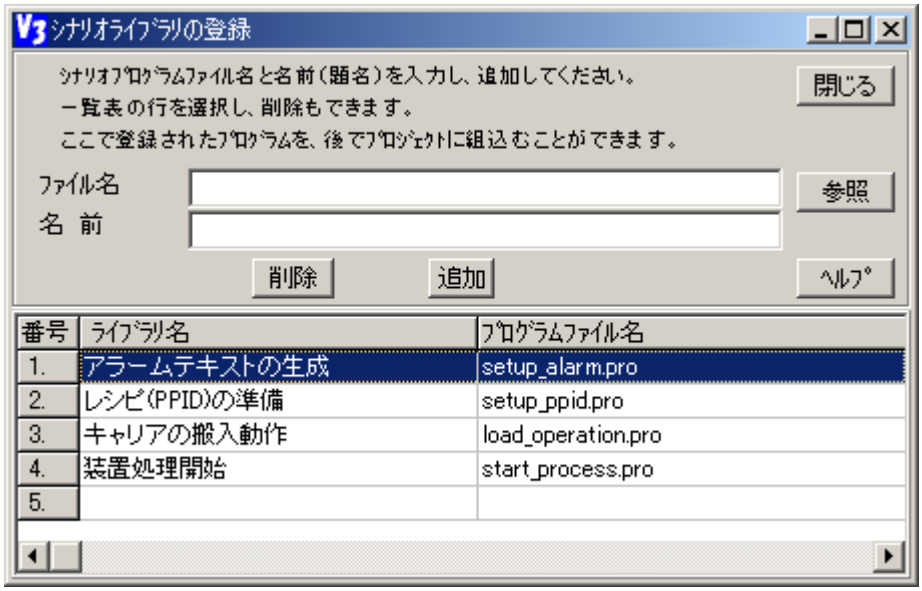
⑤ EstComm 作成が終ったら、次にこのプログラムをライブラリに登録することにします。

（2）プログラムファイルのライブラリ登録操作

① ﾂｰﾙ(T)ﾒﾆｭｰのﾌﾟﾛｸﾞﾗﾑのｼﾅﾘｵﾗｲﾌﾞﾗﾘ登録(S)タブをクリックします。

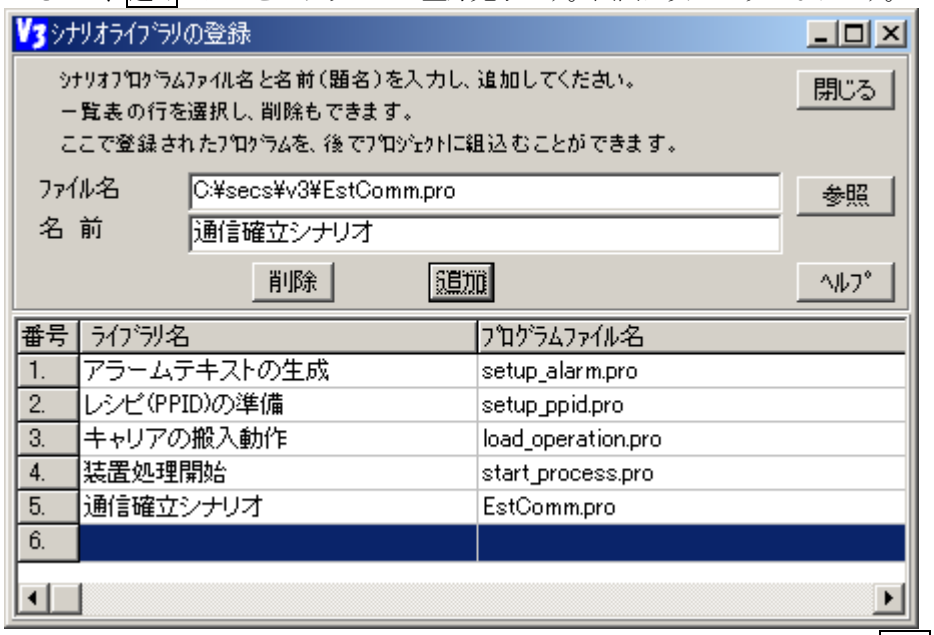


例えば、次のような登録画面が表示されます。



(注) この画面では、既に 4 つのｼﾅﾘｵが登録されていることにします。
ﾍﾙﾌﾟﾎﾞﾀﾝで操作の詳細が分かります。

② ここで、名前の欄に「通信確立シナリオ」と入力ます。
ﾌｧｲﾙ名は、参照ボタンを使って、ファイルを開く画面で、（1）で作成した EstComm.pro を選択し、設定します。
そして、追加ボタンをクリックして登録完了です。画面は次のようになります。



一旦登録したものを削除したい場合には、削除したいライブラリ名を選択し、削除ボタンをクリックし

てください。

③ 登録操作は閉じるボタンのクリックで終了します。

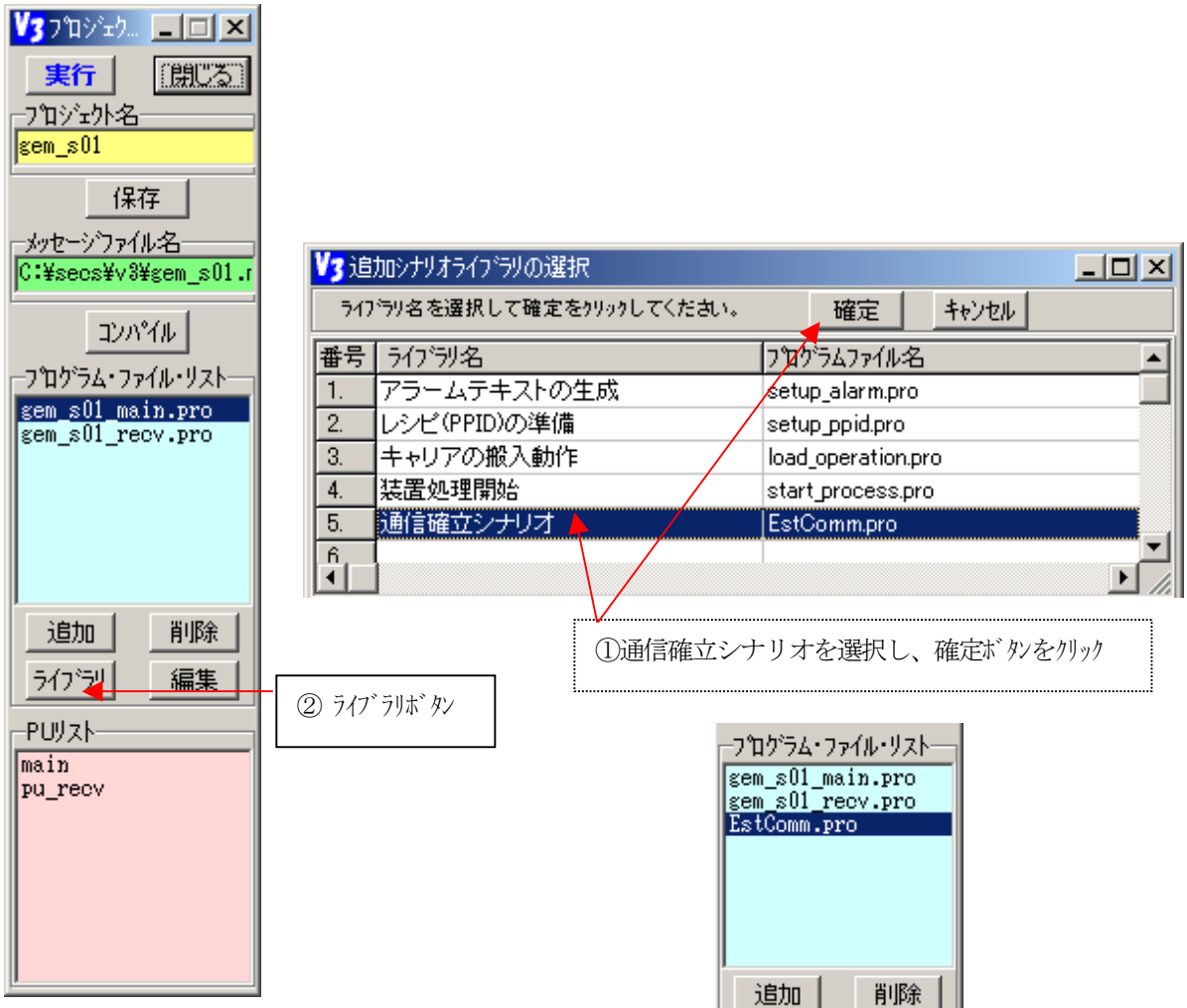
（3）シナリオライブラリプログラムの組み込み

たとえば、gem_s01.MSG ファイルから自動生成された gem_s01.prj プロジェクトの gem_s01_main.pro の中から EstComm.pro を起動し、通信確立のためのプログラミングと操作について説明します。

① 最初に gem_s01.msg を使ったプロジェクトを自動生成します。

② ライブラリの中から EstComm.pro をプロジェクトに追加するため、ﾗｲﾌﾞﾗﾘボタンをクリックします。

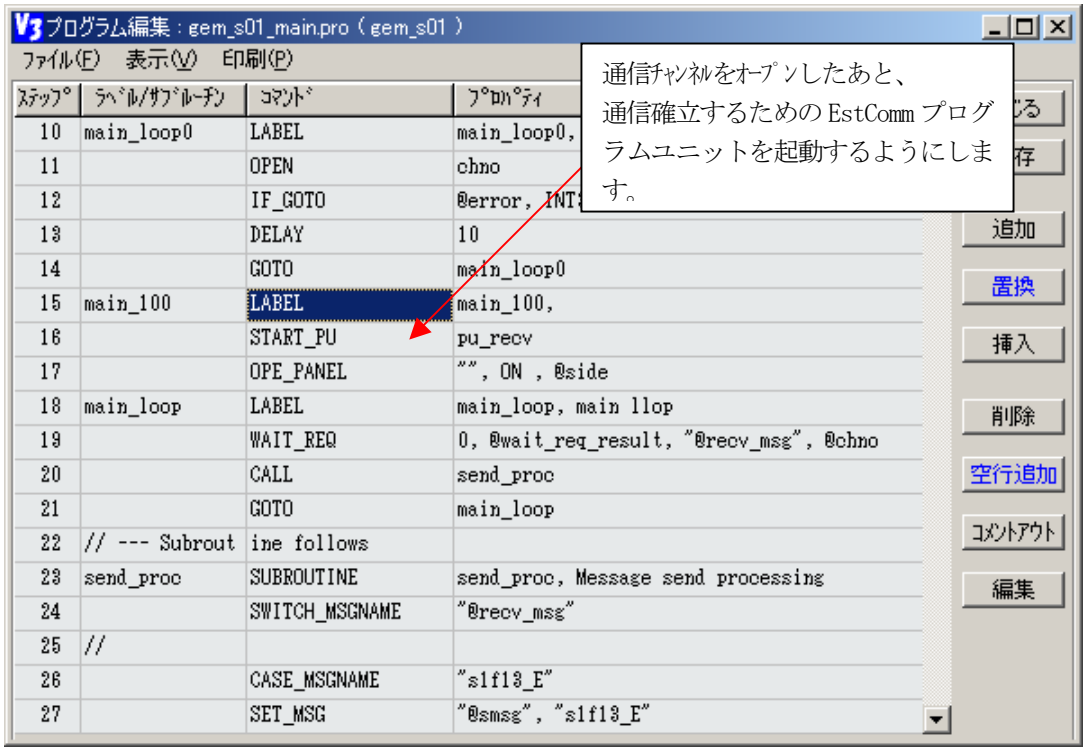
③ 表示されたライブラリ画面から EstComm を選択し、確定ボタンをクリックして追加します。



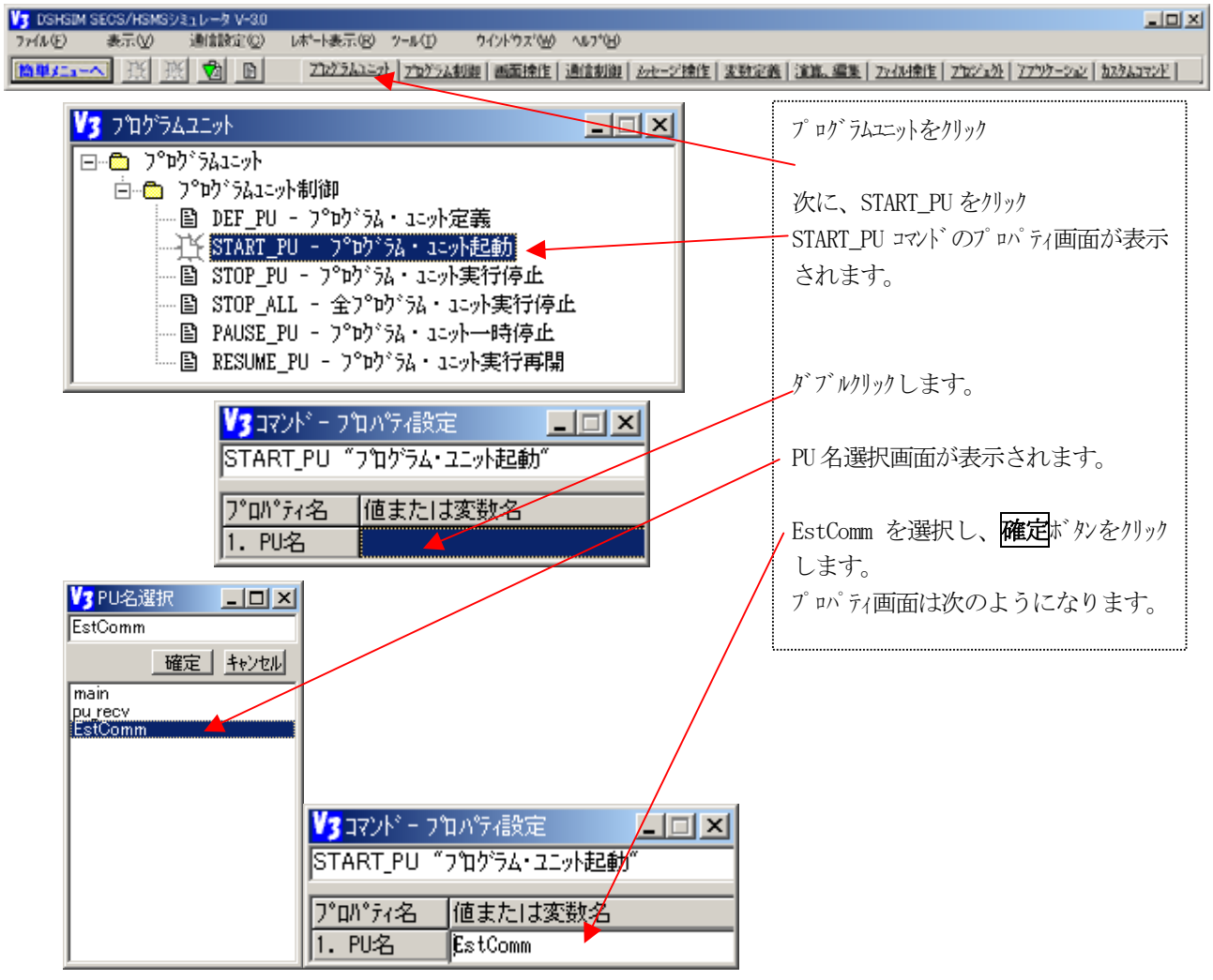
結果、プロジェクト構成画面は右図のようになります。

プログラムファイルとして、EstComm.pro が追加され、PU EstComm がプロジェクトに加えられました。

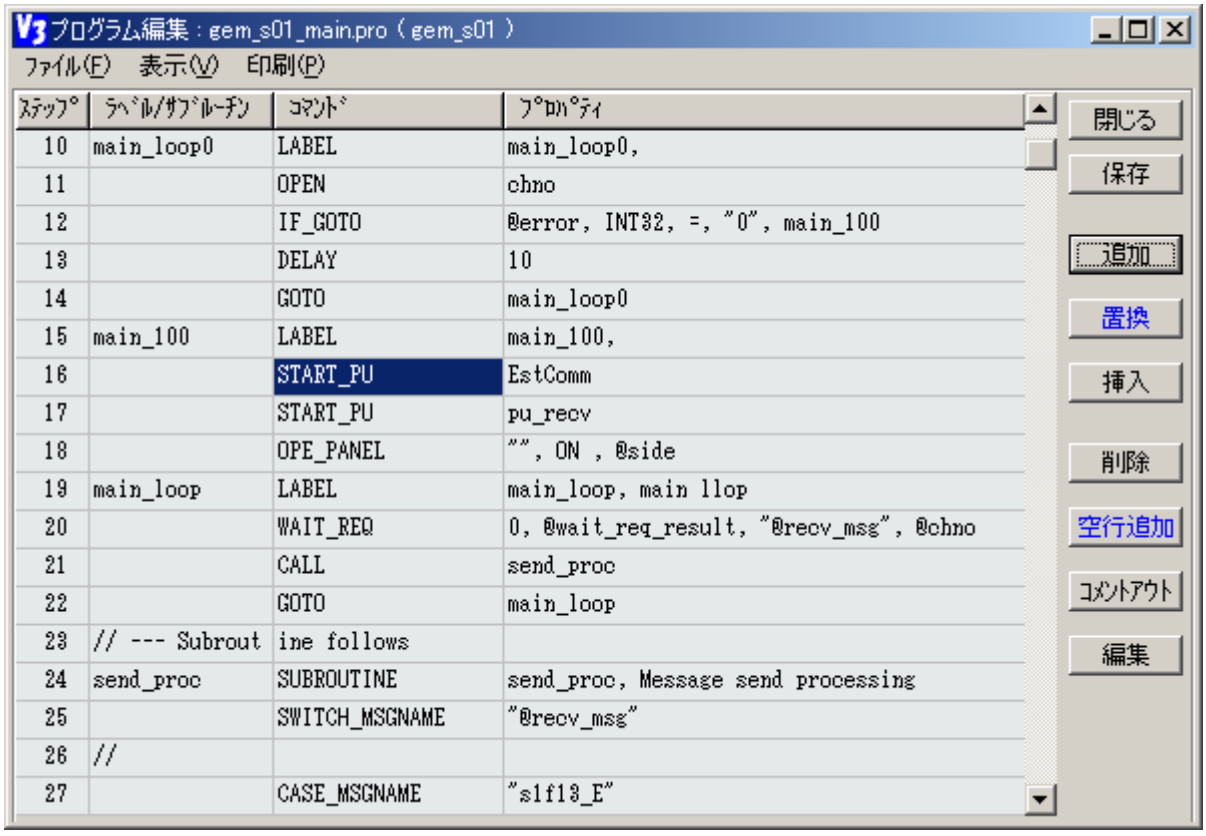
④ 引き続き、gem_s01_main.pro に PU　EstComm を起動するためのコマンドを追加します。



main_100 を選択し、START_PU コマンドを追加するために、

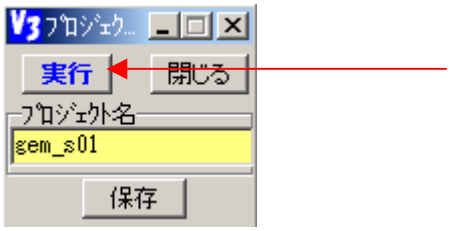


結果 START_PU EstComm が追加されます。



以上の操作でライブラリプログラム EstComm.Pro をプロジェクト gem_s01.prj に追加することができました。

このあと、実行ボタンのクリックでプロジェクトを実行することができ、gem_s01_main.pro の main PU の実行が開始され、チャンネルのオープン後、先ほど追加した"STRT_PU EstComm" コマンドで、EstComm.pro が実行され、S1F13 メッセージが相手装置に送信されることになります。



4． プロジェクトの実行操作

プロジェクトを実行する（通信を行う）ための操作について記述します。

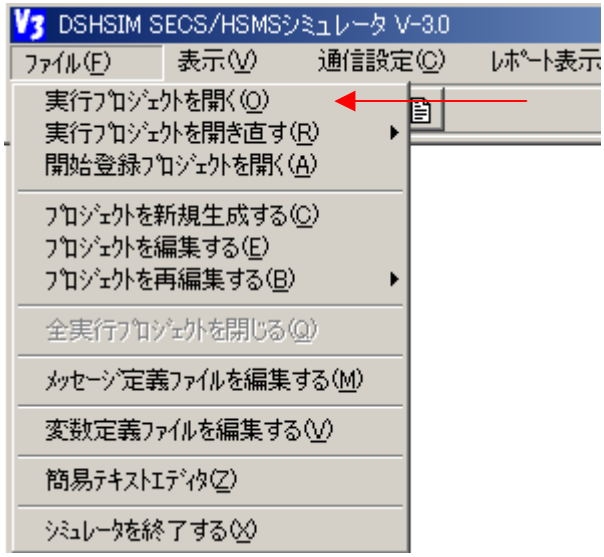
4．1 実行プロジェクトを開く

（1）プロジェクトを開く

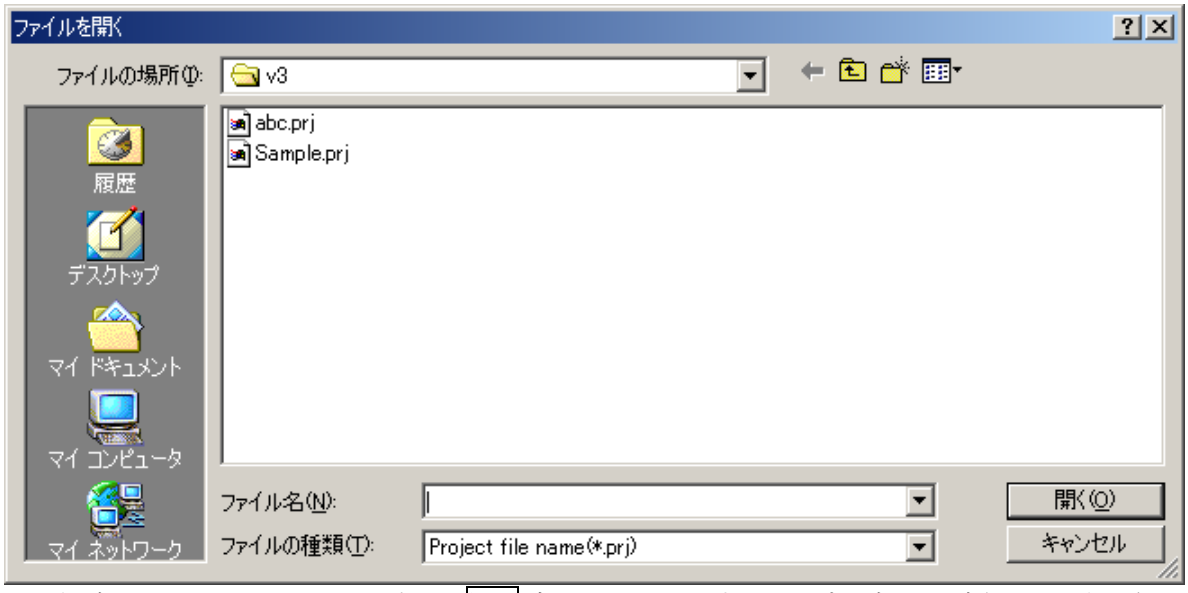
プロジェクトの生成、編集で作成されたプロジェクトの実行するために、まず、実行プロジェクトを開きます。開くための操作には以下の方法があります。

① 簡単メニューから開始します。

② ﾌｧｲﾙ(F)メニューの実行ﾌﾟﾛｸﾞﾗﾑを開く(O)タブのクリックで行います。

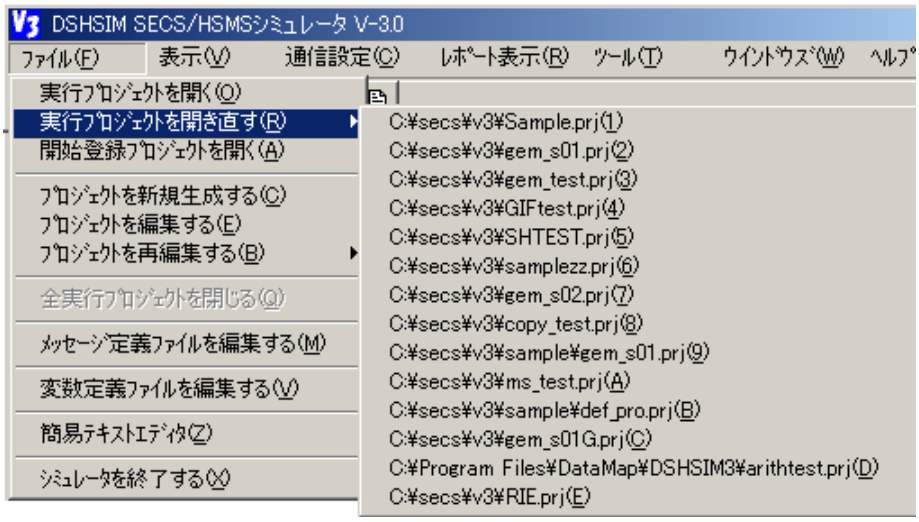


クリックすると、ファイルを開くダイアログ画面が表示されるので、そこで、開きたいプロジェクトファイルを選択し、開きます。



たとえば、ここで、Sample.prj を選択し、開く ボタンをクリックすると、プロジェクト実行画面が表示されます。

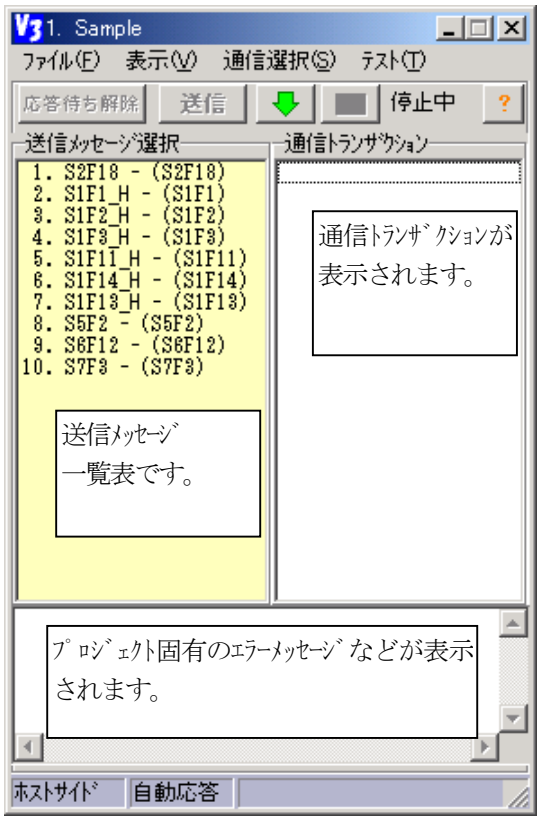
③ ﾌｧｲﾙ(F) メニューの**実行ﾌﾟﾛｸﾞﾗﾑを開きなおす(R)** タブによって表示される履歴リストから選択します。
履歴リストは、次のような画面が表示されます。ここで、開きたいプロジェクトファイルを選択します。



④ **ツールボタンの**  **の**クリックで開きます。クリックの後は、②の操作と同じです。

⑤　開始プログラム登録画面からの実行
開始プロジェクトの登録と実行操作を参照ください。

プロジェクトが正常に開かれると、次のプロジェクト実行パネル画面が表示されます。



開く途中でプロジェクトとそれを構成するプログラムファイル上にエラーを検出した場合
シミュレータは、エラーの内容を処理状況画面に表示し、開く動作を中止します。ユーザは、エラーの内容を確認し、訂正した上でもう１度やり直してください。

（2）プロジェクトを実行開始する

プロジェクトの実行開始は**ﾌｧｲﾙ(F)**ﾒﾆｭｰの**実行開始する(R)**タブのクリック、または、  （実行開始）ボタンのクリックで行います。
以下、開始した後の操作について、自動生成されたプロジェクト Sample.prj を例に、手順を説明します。
（ユーザ作成プロジェクトについては、実行開始後、組み込まれたプログラムによって操作が決定されます。）

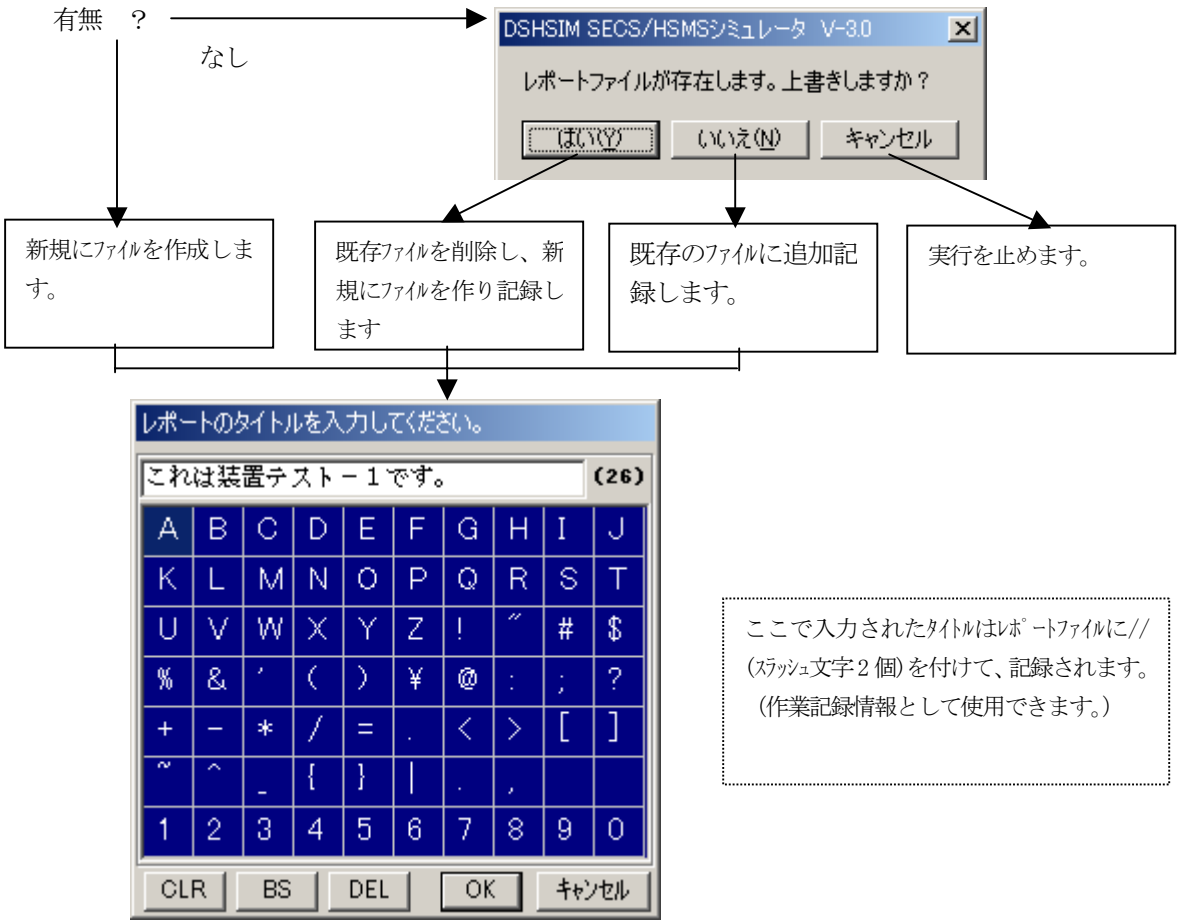
① 最初に、通信チャンネル番号の選択画面が表示されます。
プロジェクトの新規作成でデフォルトで設定したチャンネル番号がチェックされて表示されます。

自動生成時に、“ﾁｬﾝﾈﾙを問い合わせない“を選択して生成された場合は、本問合せ画面は表示されません。

詳細ボタンのクリックで選択されているチャンネルの詳細情報を見ることができます。

確定ボタンのクリックで次のレポートファイルの入力に進みます。

② 実行プロジェクトのレポートファイルに関する操作を行います。
シミュレータは、既にレポートファイルが存在しているかどうかを調べ、有無によって以下のような画面を表示します。操作の流れは、次のようになります。



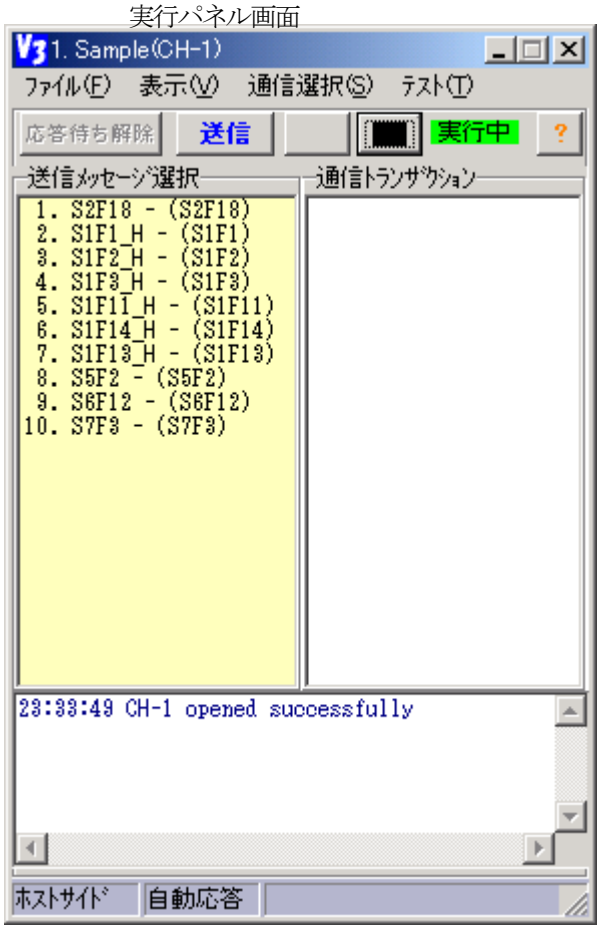
ここで入力されたﾀｲﾄﾙはﾚﾎﾟｰﾄﾌｧｲﾙに//(ｽﾗｯｼｭ文字２個)を付けて、記録されます。
（作業記録情報として使用できます。）

ここで、OKボタンをクリックするとプロジェクトの実行が開始され、処理状況画面には、オープンされたチャンネル情報が表示され、実行パネルの画面にも送信メッセージの一覧表示が現れます。

処理状況画面(ﾛｸﾞ画面)

```
08.18 15:07:48 *- [ CH=2 HSMS ｵｰﾌﾟﾝ情報 ] ----------------*
08.18 15:07:48     Session    : SS
08.18 15:07:49     Mode       : PASSIVE
08.18 15:07:49     SessionID  : 1111
08.18 15:07:49     Port       : 5002
08.18 15:07:49     T3         :  450 x 100 ms
08.18 15:07:49     T5         :  100 x 100 ms
08.18 15:07:49     T6         :   50 x 100 ms
08.18 15:07:49     T7         :  100 x 100 ms
08.18 15:07:49     T8         :   50 x 100 ms
08.18 15:07:49     LINKTEST   : 15 sec
08.18 15:07:49     WBLK_CK    : OFF
08.18 15:07:49     DVID_CK    : OFF
08.18 15:07:49     S9F1       : OFF
08.18 15:07:49     S9F9       : OFF
08.18 15:07:49     S9F11      : OFF
08.18 15:07:49 *----------------------------------------*
08.18 15:07:49     CH-2 now listening
```

実行パネル画面



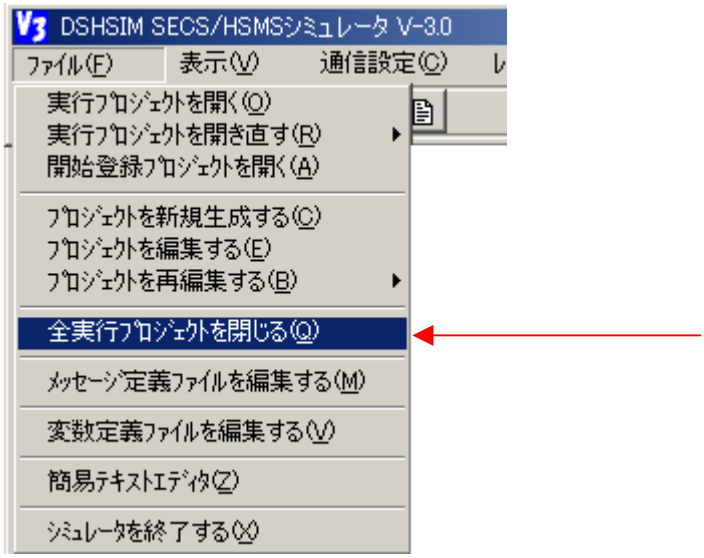
実行中の下の緑の横バー ▬ は HSMS 通信の場合、Selection が確立していない間、点滅表示します。この表示が消えることは Selection が確立し、メッセージの送信が可能であることを意味しています。

送信ボタンは、Selection が確立しないと有効にならないので、Selection 未確立状態では送信できません。

実行パネルの操作とメッセージ送信については、実行パネルの操作を参照ください。

（3）全実行プロジェクトを閉じる

ファイル(F)メニューの**全実行プロジェクトを閉じる(Q)**タブのクリックによって、開かれている全実行プロジェクトをまとめて終了させることができます。



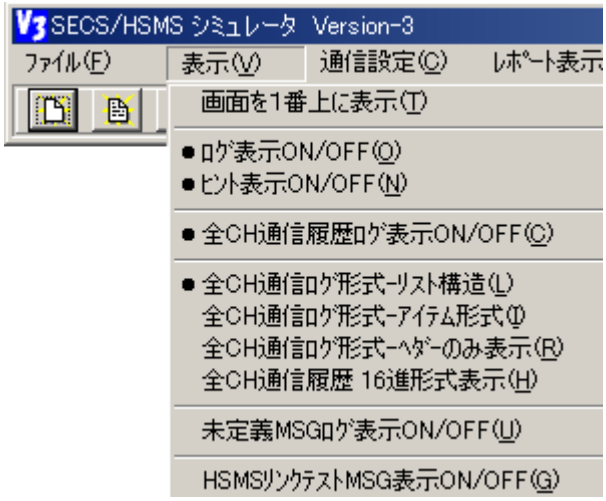
実行中のプロジェクトは、実行停止した上で、実行パネル画面が終了します。

４．２　システム全体の画面表示制御の操作

システム全体の画面情報表示の制御に関して説明します。
本操作は、プロジェクト実行時以外にも制御することができます。

（1）操作は、メイン画面の表示(V)メニューの中のタブのクリックで行います。

全タブは、チェック（●）の ON/OFF で指定するようになっています。タブ単独のものと複数のタブがグループになっているものもあります。単独のものは、クリックによってチェックの状態が反転します。
グループになっているものは、グループの中の１つだけがチェック ON(●)になります。



次ページに各メニュータブのチェックの意味一覧表を示します。

メニュータブのチェック状態の意味一覧表

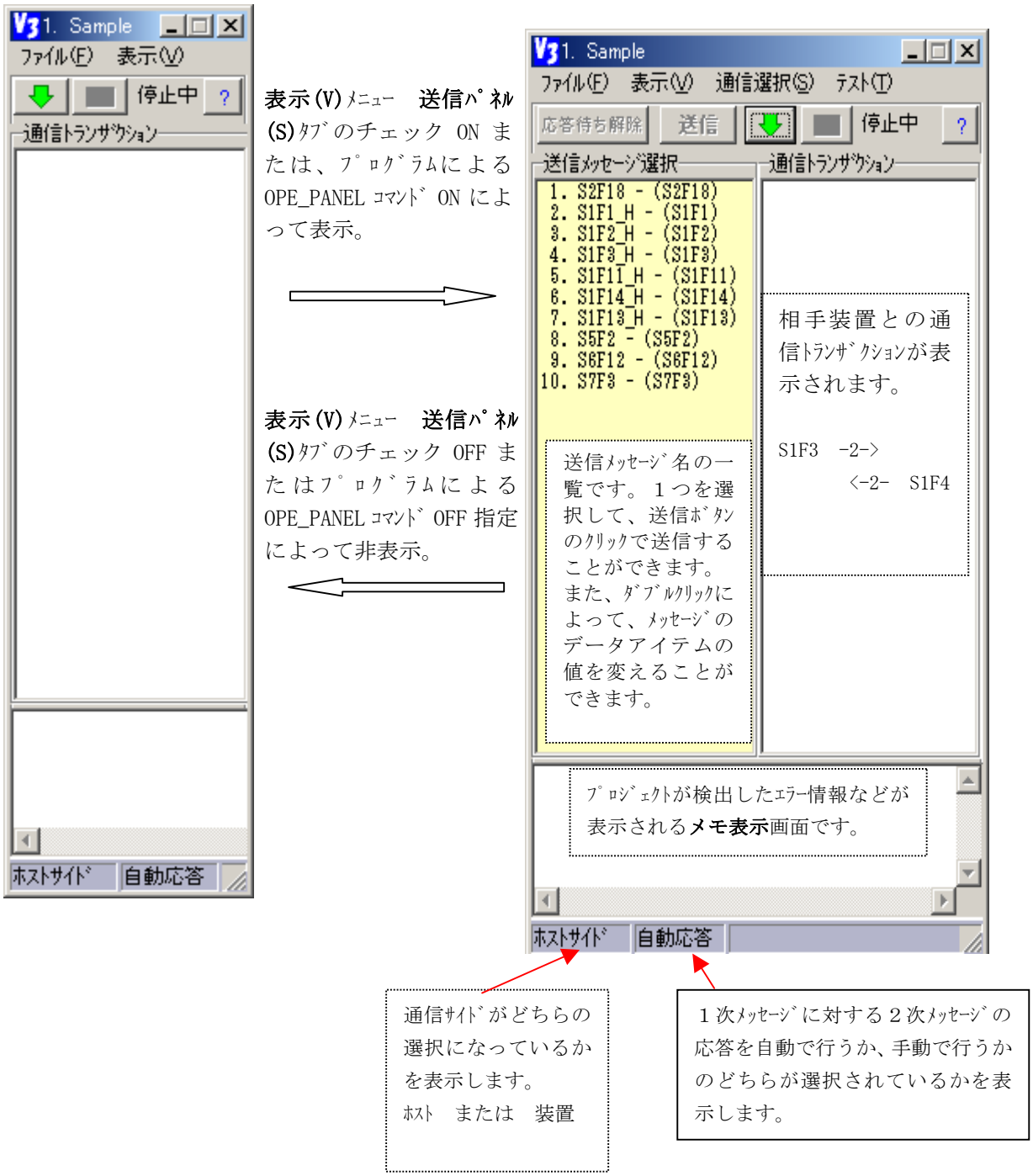
タブの表示	チェック ON 時	チェック OFF 時	注　釈
画面を１番上に表示(T)	画面を最大表示にし、MAIN 以外の画面表示の順位を１番上にし、ﾊﾞｯｸｸﾞﾗﾝﾄﾞ画面をｸﾞﾘｰﾝにします。	画面のｻｲｽﾞを通常(NROMAL)にし、ﾊﾞｯｸｸﾞﾗﾝﾄﾞ画面も元に戻し、画面表示順位も通常にします。	ｼﾐｭﾚｰﾀ起動時は通常です。 順位を 1 番にするとﾃﾞｽｸﾄｯﾌﾟの画面を見えなくすることができます。 ﾍﾙﾌﾟ表示時は通常にすることをお勧めします。
ﾛｸﾞ表示 ON/OFF(O)	ﾛｸﾞ（処理履歴画面）が表示されます。	ﾛｸﾞ（処理履歴画面）を閉じます。	
ﾋﾝﾄ表示 ON/OFF(N)	各画面のﾎﾞﾀﾝなどのｵﾌﾞｼﾞｪｸﾄﾌｫｰｶｽ時にﾋﾝﾄ表示を行います。	ﾋﾝﾄ表示を行いません。	ﾋﾝﾄは通常処理履歴画面の状態表示部（１番下）に表示されます。 通信環境定義ﾌｧｲﾙの **HINT** ｺﾏﾝﾄﾞを使って起動時の ON/OFF 選択設定ができます。
全 CH 通信履歴ﾛｸﾞ表示 ON/OFF(C)	全ﾁｬﾝﾈﾙの通信ﾒｯｾｰｼﾞを処理履歴画面に表示します。	全ﾁｬﾝﾈﾙの通信ﾒｯｾｰｼﾞを処理履歴画面に表示しないようにします。	全ﾁｬﾝﾈﾙに適用します。 （CH 個別の選択は、この後ﾌﾟﾛｼﾞｪｸﾄ・ﾊﾟﾈﾙの操作で変更できます。）
全 CH 通信ﾛｸﾞ形式-ﾘｽﾄ構造(L) 全 CH 通信ﾛｸﾞ形式-ｱｲﾃﾑ形式(I) 全 CH 通信ﾛｸﾞ形式-ﾍﾀﾞｰのみ表示(R)	ﾁｪｯｸ(=ON)されている形式で通信ﾛｸﾞを表示します。	左と同様	ｸﾞﾙｰﾌﾟであり、３つの中の１個だけが ON になります。 （CH 個別の選択は、この後ﾌﾟﾛｼﾞｪｸﾄ・ﾊﾟﾈﾙの操作で変更できます。）
全 CH 通信履歴 16 進形式表示(H)	通信ﾒｯｾｰｼﾞを 16 進形式でも表示します。	16 進表示はしません。	ﾘｽﾄ、ｱｲﾃﾑまたはﾍﾀﾞｰ表示の他に 16 進形式での表示もできます。
未定義 MSG ﾛｸﾞ表示 ON/OFF(U)	相手から未定義ﾒｯｾｰｼﾞを受信したら、その旨を処理履歴画面に表示します。	特に表示しません。	全ﾁｬﾝﾈﾙに適用されます。
HSMS ﾘﾝｸﾃｽﾄ MSG 表示 ON/OFF(G)	LINKTEST.REQ, .RSP ﾒｯｾｰｼﾞの表示を行います。 ﾛｸﾞﾌｧｲﾙにも記録します。	表示はしません。また、ﾛｸﾞﾌｧｲﾙにも記録しません。	全ﾁｬﾝﾈﾙに適用されます。 （CH 個別の選択は、この後ﾌﾟﾛｼﾞｪｸﾄ・ﾊﾟﾈﾙの操作で変更できます。）

４．３　実行パネルの操作

プロジェクト実行時に使用する実行パネル画面の操作について説明します。

（１）送信操作パネル部分の表示 ON/OFF

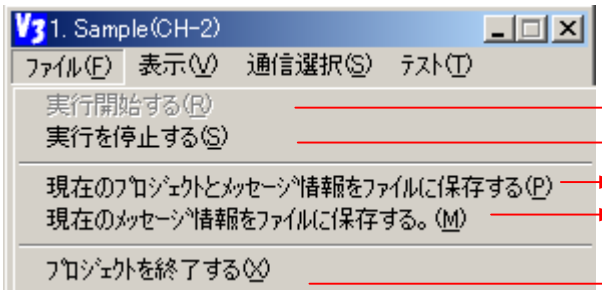
プロジェクトを開くと下の左側のパネルが表示されます。そして、下図に示すようにメッセージ送信のための画面の表示／非表示の操作を表示メニューまたはプログラムコマンドの実行によって制御できます。
画面の各部の用途の説明は右側の画面中に示します。



（2）メニューの説明

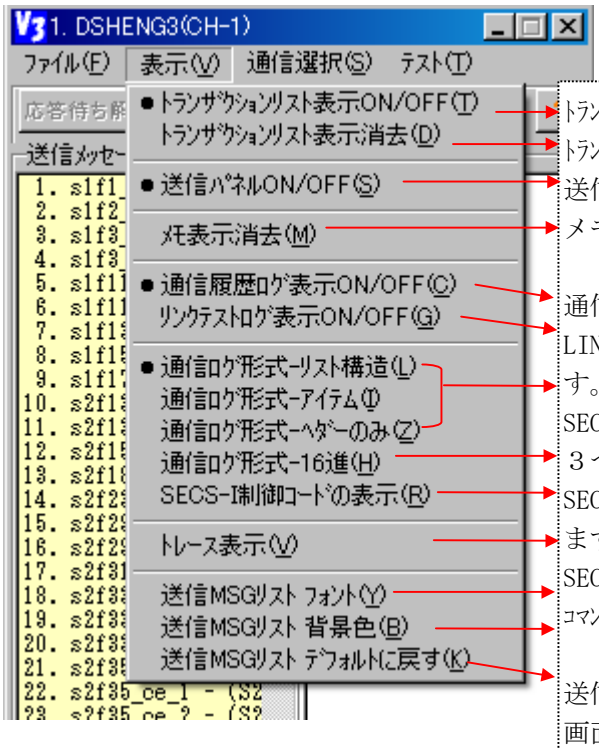
① ﾌｧｲﾙ(F)メニュー

実行開始、停止、ﾌﾟﾛｼﾞｪｸﾄの終了に使用します。また、これまでオペレータがデータアイテムを設定変更したメッセージの内容をそのままファイルに残すに使用します。



実行を開始する。
実行を停止する。
現在の状態のﾌﾟﾛｼﾞｪｸﾄ情報を別名ﾌｧｲﾙに保存
現在の状態のﾒｯｾｰｼﾞ情報を別名のﾌｧｲﾙに保存

② 表示(V)メニュー

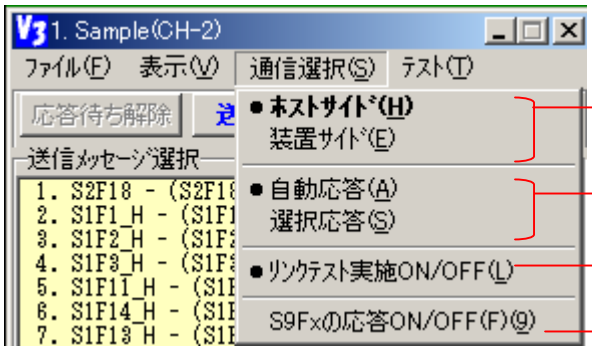


ﾄﾗﾝｻﾞｸｼｮﾝ表示を行うかどうかを選択します。
ﾄﾗﾝｻﾞｸｼｮﾝ表示内容を消去します。
送信ﾊﾟﾈﾙの表示するかどうかを選択します
メモ表示画面を H 消去します。

通信履歴ﾛｸﾞをするかしないかを選択します。
LINKTEST のﾛｸﾞ表示をするかどうかの選択します。
SECS 通信ﾒｯｾｰｼﾞのﾛｸﾞ表示形式を選択します。
３つのうちの１つを選択します。
SECS 通信ﾒｯｾｰｼﾞの内容の 16 進数表示を選択します。
SECS-I の制御ｺｰﾄﾞの表示を選択します。
ｺﾏﾝﾄﾞの実行状態をﾄﾚｰｽ表示します。

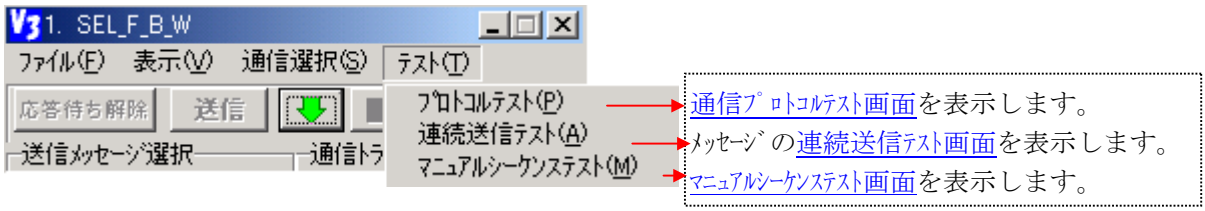
送信 MSG ﾘｽﾄ画面の文字ﾌｫﾝﾄ,ｻｲｽﾞ、色、ｽﾀｲﾙと画面

③ 通信選択(S)メニュー

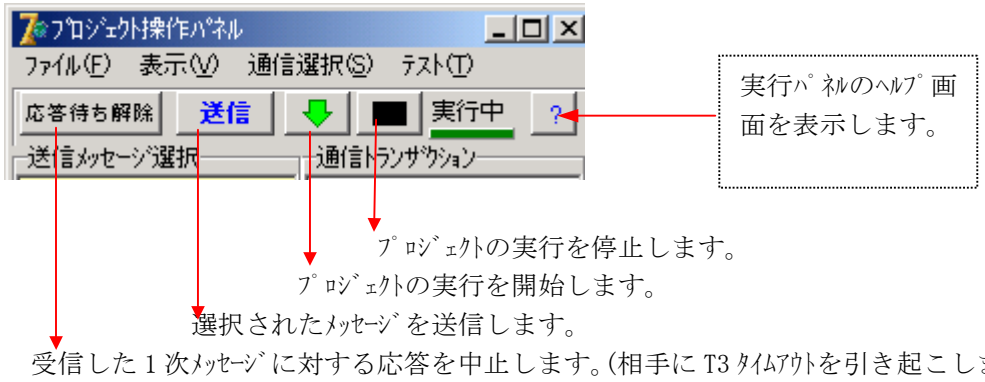


通信ｻｲﾄﾞを選択します。ｻｲﾄﾞによって送信ﾒｯｾｰｼﾞが変わります。
2 次ﾒｯｾｰｼﾞの応答を自動で行うか、ｵﾍﾟﾚｰﾀの選択で行うかを選択します。
LINKTEST.REQ を送信するかどうか選択します。
S9F1,7,9,11 の送信を行うかどうかを選択します。

④ テスト(T)メニュー

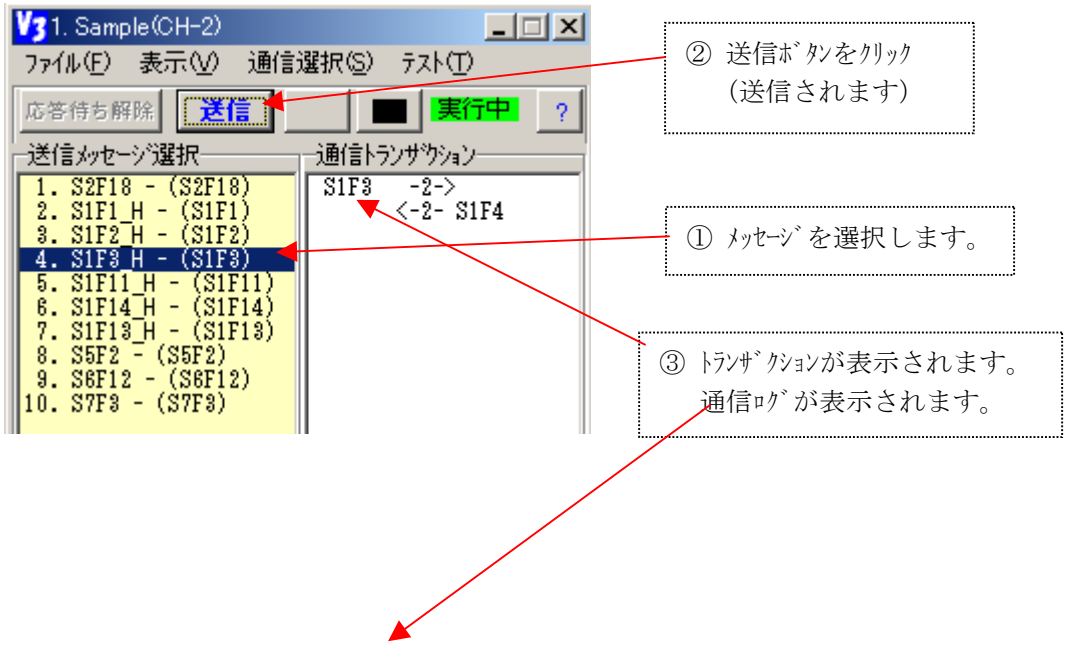


（3）操作ボタンの説明



（4）メッセージの送信

メッセージの送信は、送信メッセージリストの中の１つのメッセージをマウスのクリックで選択し、送信ボタンのクリックで送信します。送信メッセージと受信応答メッセージは処理状況画面にログ表示され、または環境定義ファイルで指定されたログファイルにも記録されます。そして、レポートファイルにも記録されます。操作と動作の順番は下図の説明ボックスのとおりです。

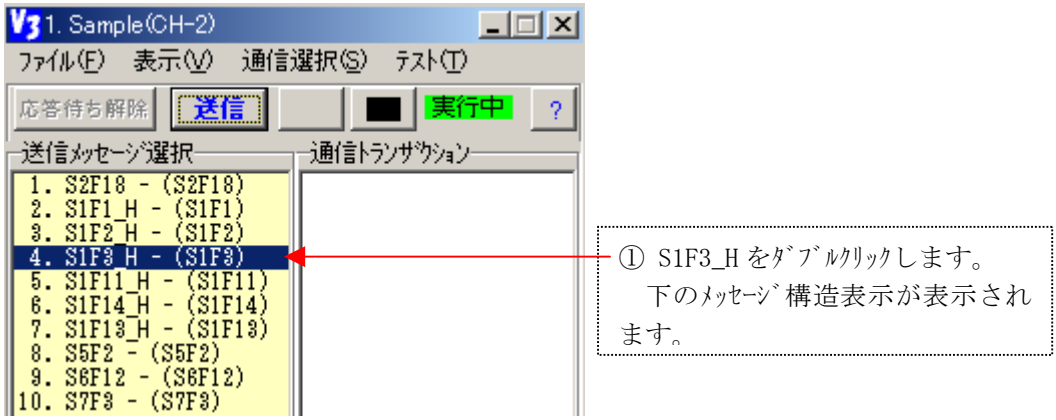


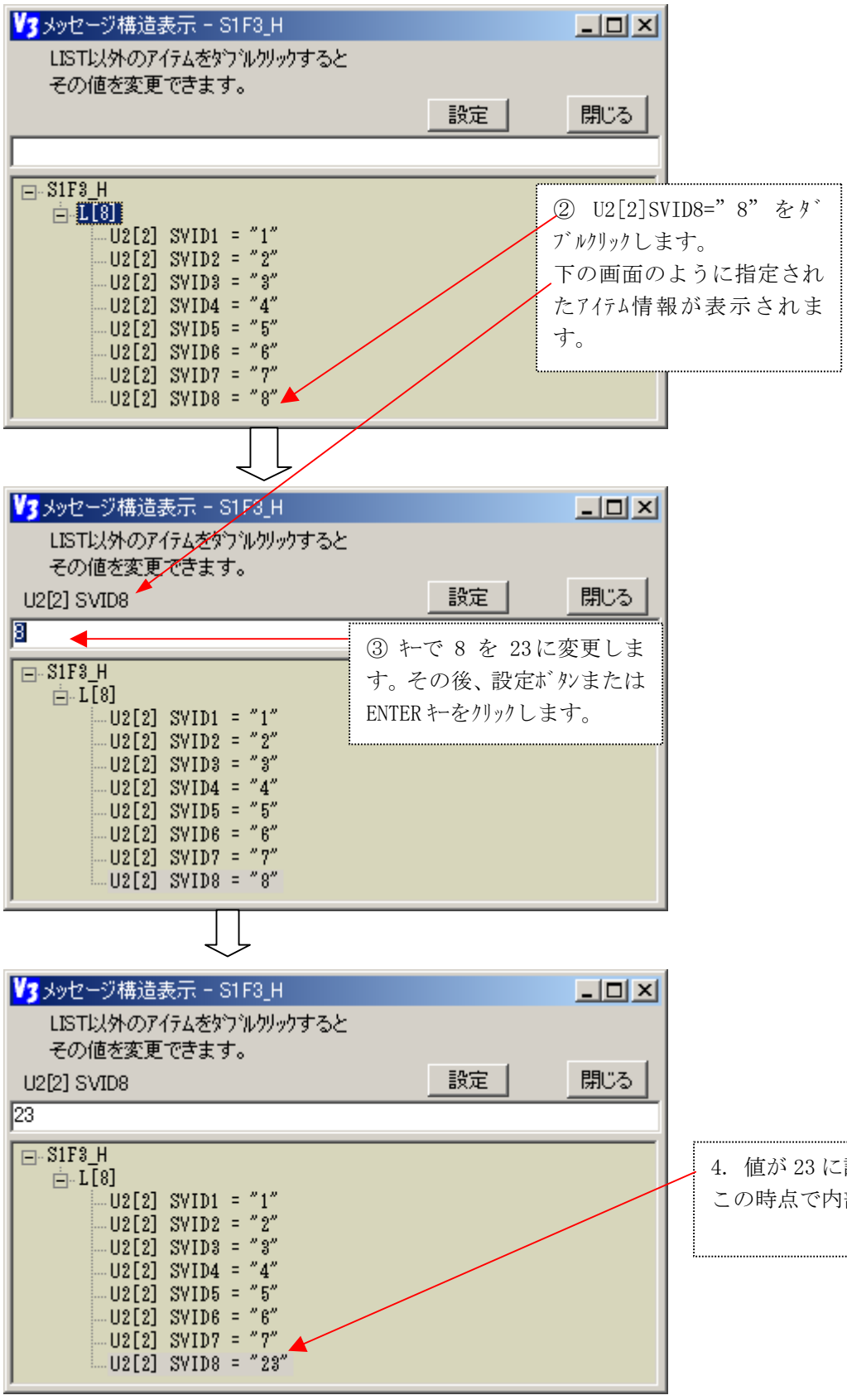
（5）送信メッセージのデータ変更

メッセージを送信する前に、そのメッセージに含まれるデータを設定変更して送信することができます。

たとえば、S1F3_H に含まれる SVID8 の値を 23 に変更したい場合は、次のように行います。

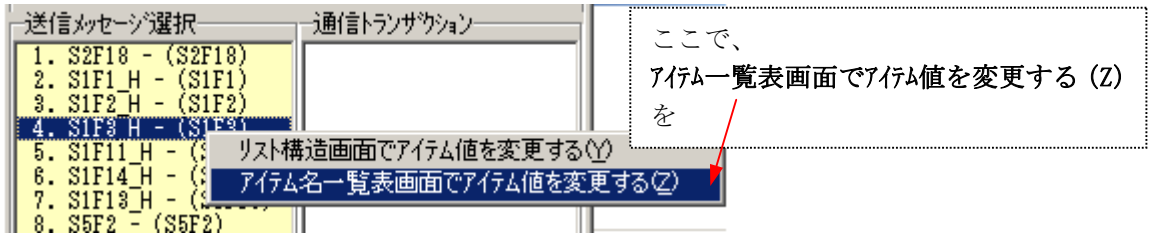
最終的に設定したメッセージ情報を通常のメッセージ定義ファイルの形式で、ファイルに保存することができます。　（１１）を参照してください。



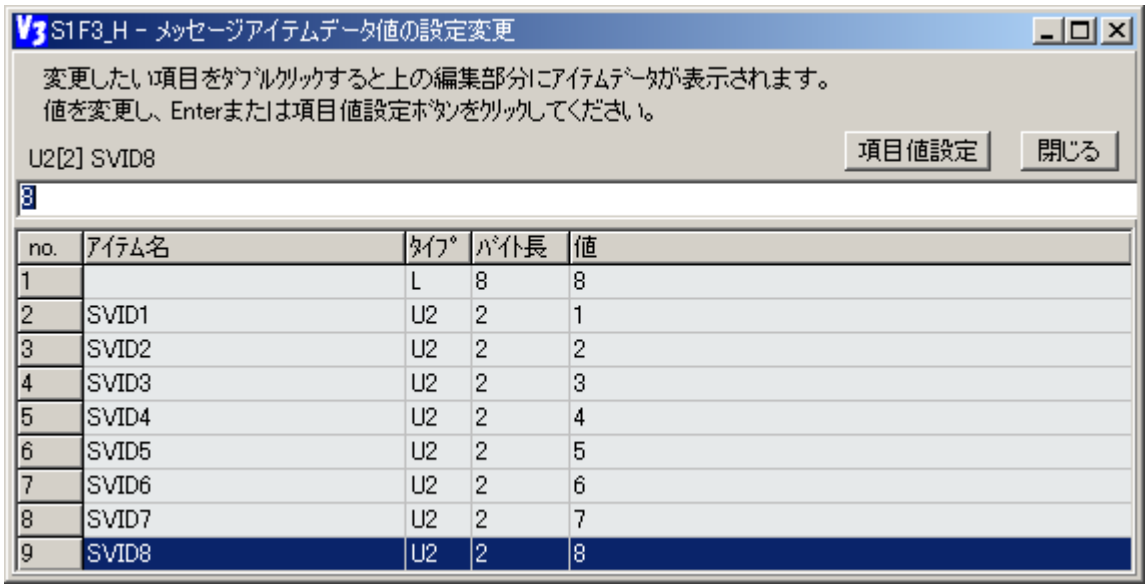


これで、変更完了です。この後、S1F3_H を送信すると、SVID8=23 のメッセージが送信されます。

別の方法として、送信メッセージ名を**右クリック**すると、つぎのようなポップアップメニューが出ます。



次のように全アイテムが一覧表に表示されます。変更したいアイテムをクリックして値を変更することができます。



no.	ｱｲﾃﾑ名	ﾀｲﾌﾟ	ﾊﾞｲﾄ長	値
1		L	8	8
2	SVID1	U2	2	1
3	SVID2	U2	2	2
4	SVID3	U2	2	3
5	SVID4	U2	2	4
6	SVID5	U2	2	5
7	SVID6	U2	2	6
8	SVID7	U2	2	7
9	SVID8	U2	2	8

（6）通信サイドの切り替え

通信サイドの切り替えは、（２）－③で説明した**通信選択(S)**メニューの**ﾎｽﾄｻｲﾄﾞ(H)**と**装置ｻｲﾄﾞ(E)**タブのクリックで行います。
ホスト側が選択されているときは、プロジェクトが有する通信定義メッセージのホスト側が送信するメッセージ名だけが、送信メッセージリスト画面に表示される。装置側が選択されているときは、装置側送信のメッセージ名が表示される。
切り替えは、プロジェクト生成時に両サイドの生成が選択されたプロジェクトについてのみ有効です。

（7）応答モードの切り替え

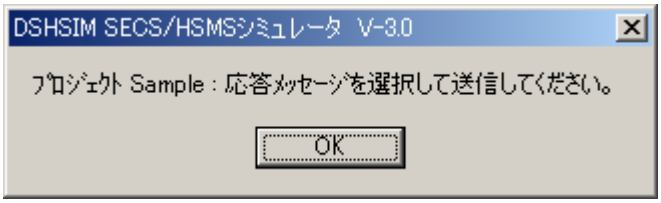
応答モードには、自動応答モードと選択応答モードがあり、切り替えは、**通信選択(S)**メニューの**自動応答(A)**と**選択応答(S)**タブのクリックで行います。

自動応答モードでは２次メッセージを期待する１次メッセージを受信した際、シミュレータは、メッセージ定義ファイルで定義されたメッセージの中に応答すべきメッセージがあるかどうかを調べます。調べた結果、応答メッセージが１個だけ見つかった場合は、そのメッセージを自動的に応答します。ただし、２個以上見つかった場合には、応答メッセージの可能性のあるメッセージ名リストが表示された画面がポップアップされます。そして、その中からオペレータが選択し、送信することになります。

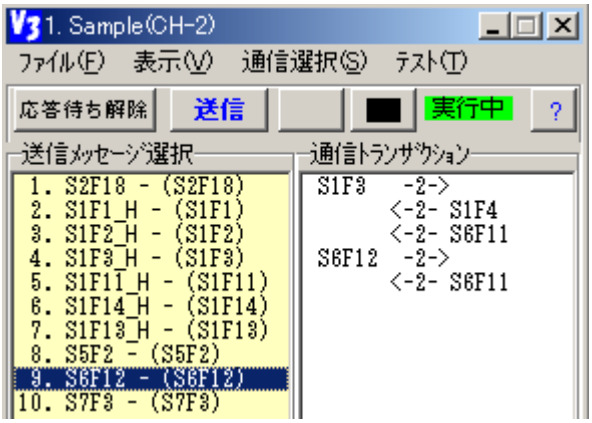
選択応答モードでは、２次メッセージを期待する１次メッセージを受信した場合、常に、オペレータにその旨を画面でオペレータに知らせます。その後、オペレータは、応答メ

ッセージを選択、送信することになります。

応答２次メッセージの選択をオペレータに求める際、画面はたとえば、次のようになります。



ＯＫボタンのクリックで、パネル表示の応答待ち解除ボタンが有効になり、応答メッセージ候補があれば、その送信メッセージ名が選択されます。



このまま、選択されているﾒｯｾｰｼﾞを送信する場合は、送信ﾎﾞﾀﾝのｸﾘｯｸで応答送信します。
（送信後、応答待ち解除ﾎﾞﾀﾝは有効でなくなります。）

また、応答待ち解除ﾎﾞﾀﾝをｸﾘｯｸすると、何も応答しないことになります。この場合、相手側では T3 ﾀｲﾑｱｳﾄが発生することにな

（８）リンクテストの実施 ON/OFF

ＨＳＭＳプロトコルでは、通信環境定義ファイルの中の各チャンネル定義の中で、リンクテストを何秒間隔で行うかを LINKTEST コマンドで指定します。この値が 0 であればリンクテストは行われません。
リンクテストは、ＨＳＭＳプロトコル上の Selection 確立した後、実施することになります。
通信選択ﾒﾆｭｰ(S)の**ﾘﾝｸﾃｽﾄ実施 ON/OFF(L)**タブのチエック OFF によって、リンクテストの実施を強制的に行わないようにできます。
リンクテストメッセージの通信ログの表示 ON/OFF の切り替えは、**表示メニュー(V)**の**ﾘﾝｸﾃｽﾄﾛｸﾞ表示 ON/OFF(G)**タブのクリックで行います。

（9）通信テスト

① ﾃｽﾄ(T)ﾒﾆｭｰのﾌﾟﾛﾄｺﾙﾃｽﾄ(P)タブのクリックによって、プロトコルテストを実行できます。
プロトコルテストは、通信チャンネルに故意にプロトコル上でエラーになる要因を作り、相手装置の通信がプロトコル上、正しく実行しているかどうかをテストするための機能です。
詳しくは、通信プロトコルのテストを参照してください。

② ﾃｽﾄ(T)ﾒﾆｭｰの**連続送信ﾃｽﾄ(A)**タブのクリックによって、連続送信テストを実行できます。
複数のメッセージを一定間隔で連続送信し、相手装置の動作を確認することができます。（応答メッセージが返ってくるかどうか、メモリリークが発生していないかどうかなどのテスト）
詳しくは、連続送信テストを参照してください。

③ ﾏﾆｭｱﾙｼｰｹﾝｽﾃｽﾄ(M)タブのクリックによって、マニュアルシーケンステストを実行できます。
予めシナリオに従って並べられている順番に適当な遅延時間をとりながらメッセージを送信させる機能です。
詳しくは、マニュアルシーケンステストを参照してください。

（10）実行プログラムのトレース表示

表示メニューの中の**ﾄﾚｰｽ表示**のチェックを ON にすることによって、現在プロジェクトが実行しているプログラムのステップ位置とコマンド名をログ表示させることができます。
プログラムの実行位置を突き止めたい場合に使用すると便利です。
下に、トレース表示例を示します。

```
09.21 10:34:41 |___ step =    3  LABEL  prj=Sample2  pu=pu_recv _______________
09.21 10:34:41 |___ step =    4  CH_CHECK  prj=Sample2  pu=pu_recv _____________
09.21 10:34:41 |___ step =    5  IF_GOTO  prj=Sample2  pu=pu_recv _____________
09.21 10:34:41 |___ step =    6  DELAY  prj=Sample2  pu=pu_recv _______________
09.21 10:34:41 |___ step =    7  GOTO  prj=Sample2  pu=pu_recv ________________
09.21 10:34:41 |___ step =    3  LABEL  prj=Sample2  pu=pu_recv _______________
09.21 10:34:41 |___ step =    4  CH_CHECK  prj=Sample2  pu=pu_recv _____________
09.21 10:34:41 |___ step =    5  IF_GOTO  prj=Sample2  pu=pu_recv _____________
09.21 10:34:41 |___ step =    6  DELAY  prj=Sample2  pu=pu_recv _______________
```

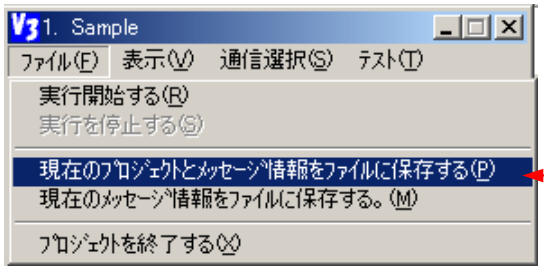
（11）現状のプロジェクトとメッセージ情報を別名ファイルに保存する。

操作の中でメッセージのデータアイテム値を変更したりしますが、その変更値をファイルに保存しておいて、後で、そのメッセージで再度シミュレーションすることでできます。
即ち、（５）で変更したメッセージデータの情報をそのままの値で、別ファイルに保存することができます。
方法は２つあり、１.プロジェクト単位での保存、2.メッセージ定義ファイルだけの保

存、です。

① プロジェクト単位での保存

ﾌｧｲﾙ(F)ﾒﾆｭｰの **現在のﾌﾟﾛｼﾞｪｸﾄとﾒｯｾｰｼﾞ情報をﾌｧｲﾙに保存する(P)**ﾀﾌﾞのクリックから始めます。



以下の画面が表示されますので、ここで、保存したいﾌﾟﾛｼﾞｪｸﾄﾌｧｲﾙ名とﾒｯｾｰｼﾞﾌｧｲﾙ名を入力し、確定ボタンをクリックするだけです。

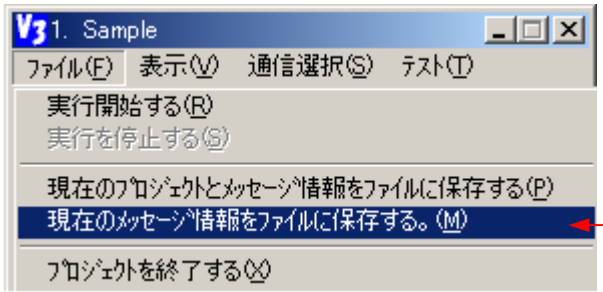
デフォルト名のまま保存したい場合は、そのまま確定をクリックしてください。

参照ボタンをｸﾘｯｸするとﾌｧｲﾙ保存用ﾀﾞｲｱﾛｸﾞ画面を使って保存ﾃﾞｨﾚｸﾄﾘも選択できます。

ここで別名で保存したプロジェクトは、すぐに実行させることができます。

② メッセージ情報だけをメッセージ定義ファイルに保存

ﾌｧｲﾙ(F)ﾒﾆｭｰの **現在のﾒｯｾｰｼﾞ情報をﾌｧｲﾙに保存する(M)**ﾀﾌﾞのクリックから始めます。



上と同様ですが、メッセージファイル名だけの入力になります。

この場合、あとで、このメッセージファイルを使用したプロジェクトを生成して実行することになります。

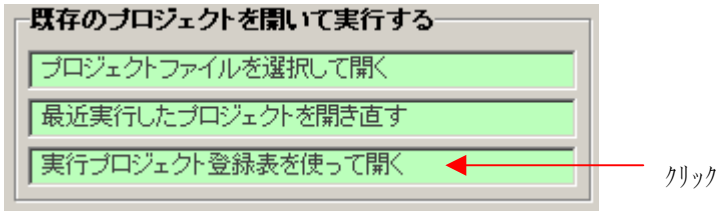
４．４　開始プロジェクトの登録と実行操作

よく使用するプロジェクトを開始プロジェクト登録ファイル(autoprj.prl)に予め登録しておき、複数のプロジェクトを少ない操作で開いて実行させることができます。たとえば、８個のプロジェクトを同時に開いて実行する場合、操作を効率よく実行させることができます。登録できるプロジェクトは最大２０個です。

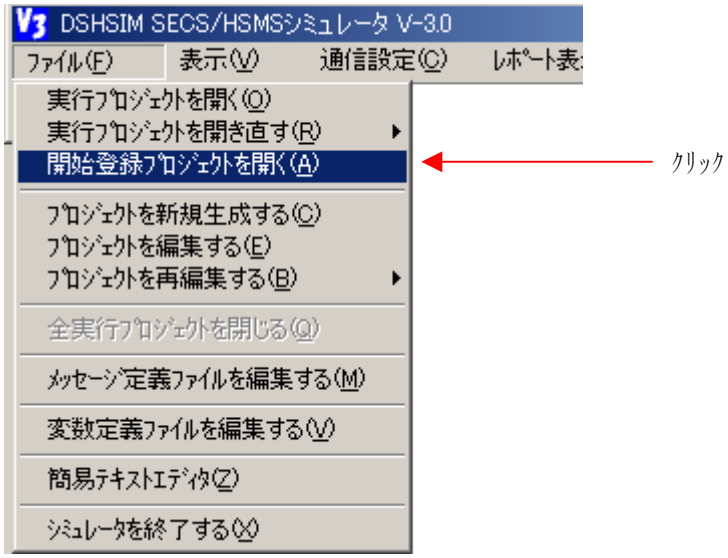
（1）登録画面とプロジェクトの登録

① 操作開始として次の２通りの方法があります。

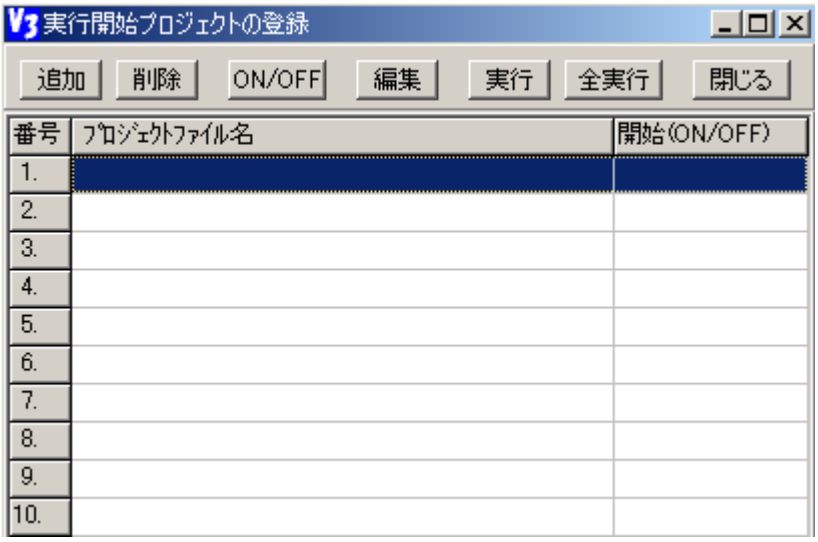
a. 簡単メニューから



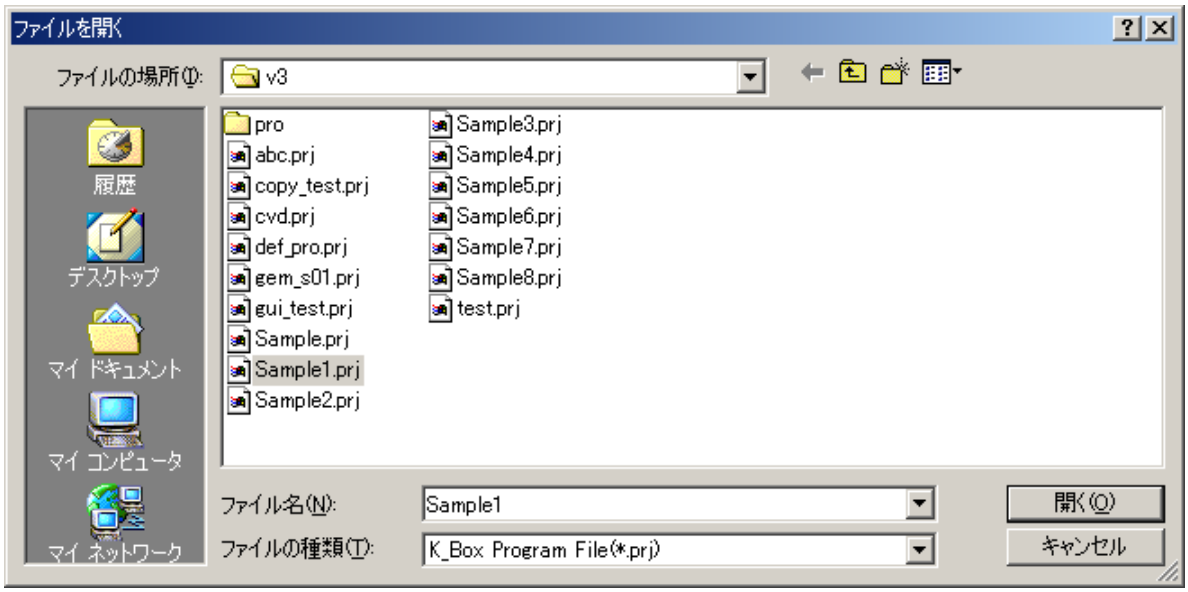
b. 登録画面は、メイン画面のﾌｧｲﾙ(F)メニュー開始登録ﾌﾟﾛｼﾞｪｸﾄを開く(A)タブのクリックによって開きます。



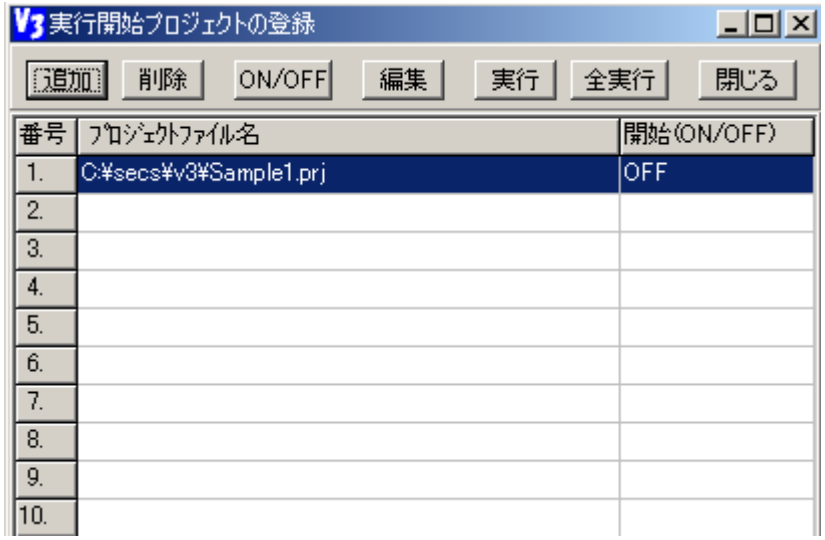
次の画面が表示されます。何も登録されていない画面です。



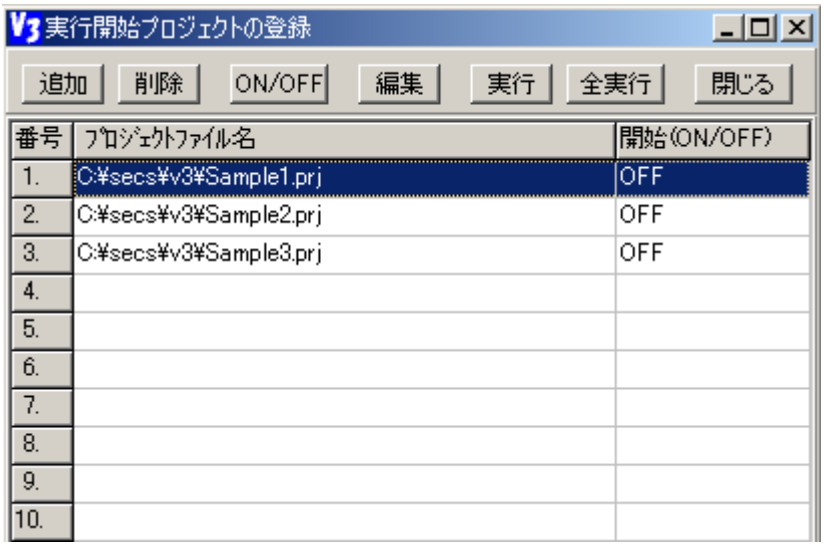
② この状態で、追加ボタンをクリックし、登録したいプロジェクトファイルを開くための画面が開かれます。
たとえば、Sample.1.prj を登録したい場合は、Sample1.prj を選択し、開くボタンをクリックします。



登録画面に、Sample1.prj が追加され、表示されます。



③ さらに、Sample2.prj , Sample3.prj を追加すると次のようになります。



（2）登録プロジェクトの実行

① 個別の実行

画面で実行したいプロジェクトを選択し、実行ボタンをクリックします。選択されたプロジェクトはただちに実行されます。

この場合、開始(ON/OFF)状態は OFF でも構いません。

② 開始 ON 状態のプロジェクトすべての実行

全実行ボタンのクリックによって、開始(ON/OFF)が **ON になっているプロジェクトを一度にまとめて実行させる**ことができます。

開始 ON にするためには、ON にしたいプロジェクトを選択し、ダブルクリックまたは、**ON/OFF** ボタンをクリックすることによって行います。状態は、クリック操作ごとに OFF<-->ON の状態が交互に切り替わります。

なお、同時に実行できるプロジェクトは、最大８個です。（システムで同時開始できるプロジェクト数が最大８個です。）

それから、開始は登録されている順に、先に開始されたプロジェクトの実行が済んだことを確認しながら行われます。たとえば、Sample1.prj の開始後、Sample2.prj の開始は下の画面で、はい(Y)がクリックされてからになります。

通常の実行操作については、プロジェクトの実行操作を参照ください。

（3）プロジェクトの編集

選択されたプロジェクトの編集ボタンのクリックによって編集操作の画面を開くことができます。

ただし、編集画面が他のプロジェクトによって使用中の場合は、できません。

編集操作については、プロジェクトの編集、プログラムの編集を参照ください。

（4）登録の削除

登録からプロジェクトを削除したい場合は、まず、削除したいプロジェクトを選択し、その後、削除ボタンをクリックします。

４．５　通信制御パラメータの設定変更

通信チャンネルのプロトコルと必要なパラメータに対する設定変更を行います。

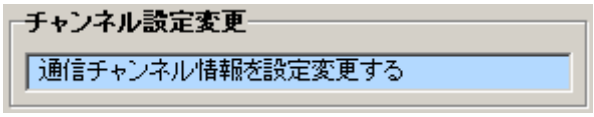
基本的には、DSHSIM.EXE シミュレータの起動パラメータで指定された通信環境定義ファイル内に指定された定義情報のパラメータの設定変更になります。（デフォルトの通信環境定義ファイル名は、COMM.DEF です。）

本操作では、プロトコルタイプ（SECS, HSHS とそのエンティティ）の変更はできません。また、変更した内容をファイルに保存することはできません。（環境定義ファイルの変更は、簡単メニュー画面で“通信環境定義ファイルを変更する”の機能で可能です。

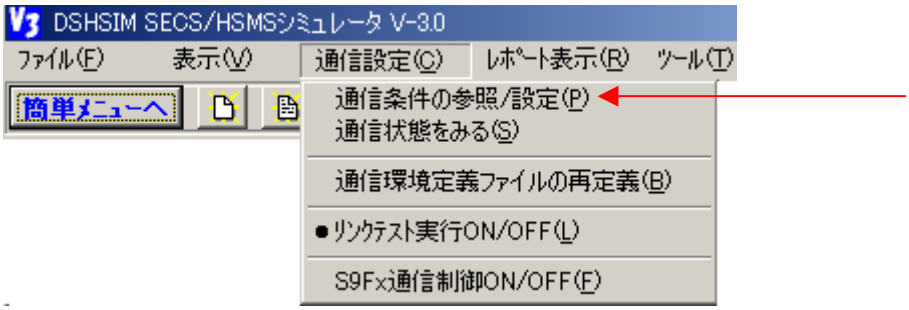
（１）操作の開始

操作は、次のどちらかの操作で始めます。

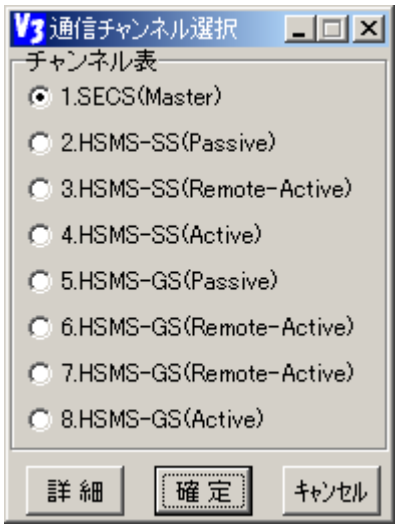
① 簡単メニュー



② メイン画面の通信設定(C)ﾒﾆｭｰの通信条件の参照/設定(P)ﾀﾌﾞ



操作を開始すると、設定変更するチャンネル選択のための画面が表示されます。

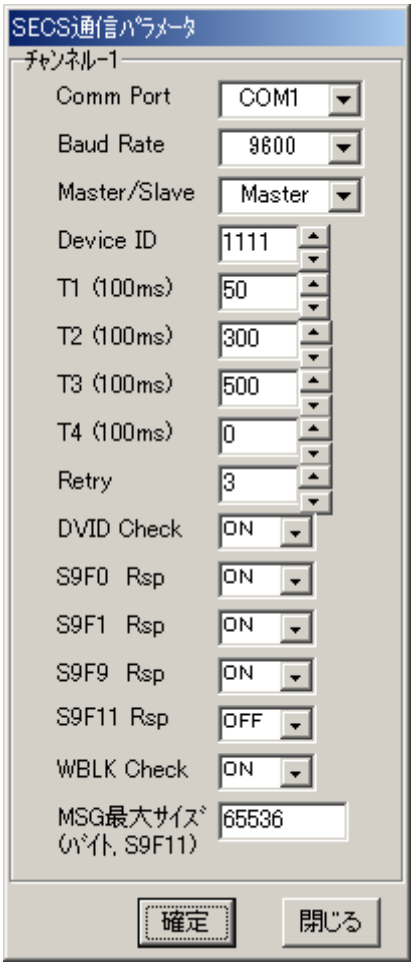


この画面で、変更したいﾁｬﾝﾈﾙを選択し、確定をｸﾘｯｸします。

以下、プロトコル別に説明します。

（2）ＳＥＣＳ－Ｉプロトコル

必要なパラメータ値を変更し、確定ボタンをクリックします。



COM1, COM2, . . ｼﾘｱﾙﾎﾟｰﾄの選択

600, 1200, 2400, 4800, 9600, 14400, 19200, 38400

MASTER/SLAVE

ﾃﾞﾊﾞｲｽ ID(16 進表現)

ﾌﾟﾛﾄｺﾙﾀｲﾏｰ

T1, T2, T3, T4

ﾌﾟﾛﾄｺﾙｴﾗｰ時のﾘﾄﾗｲ回数

ﾃﾞﾊﾞｲｽ ID の照合(ON の場合、読み捨てる)

S9F0 － BLOCK NO.ｴﾗｰ時の S9F0 応答

S9F1 － DeviceID 照合ｴﾗの S9F1 応答

S9F9 － T3 ﾀｲﾑｱｳﾄ時の S9F9 応答

S9F11－ 最大ﾒｯｾｰｼﾞ長ｵｰﾊﾞｰMSG 受信時の S9F11 応答

ﾌﾞﾛｯｸの 2 重受信ﾁｪｯｸ

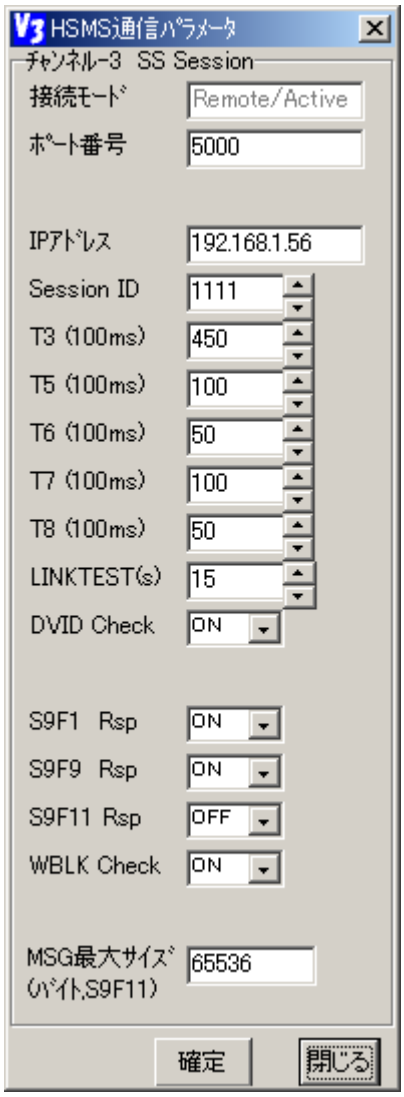
S9F11 ﾒｯｾｰｼﾞを送信するためのﾒｯｾｰｼﾞﾊﾞｲﾄｻｲｽﾞ

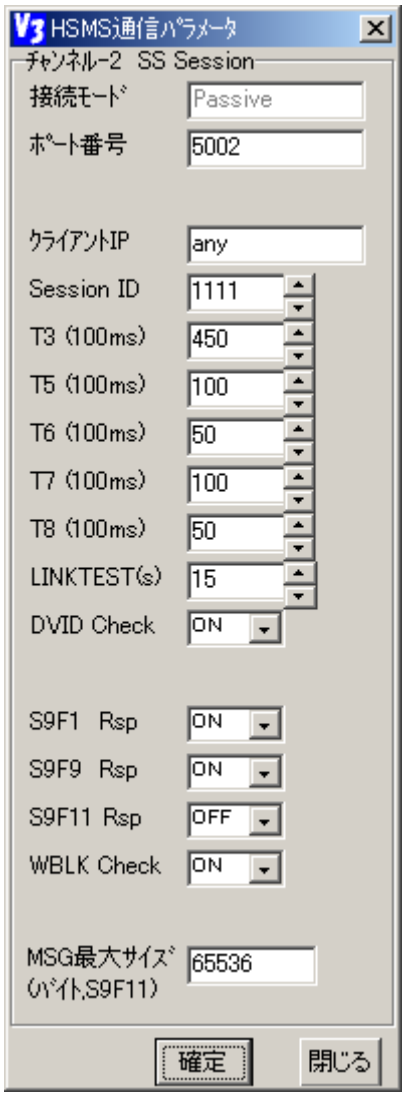
（注）Comm Port と Baud Rate は通信中は変更しても有効になりません、次に実行開始されたときに新しい値が採用されます。その他のパラメータは実行中でも変更値がただちに有効になります。

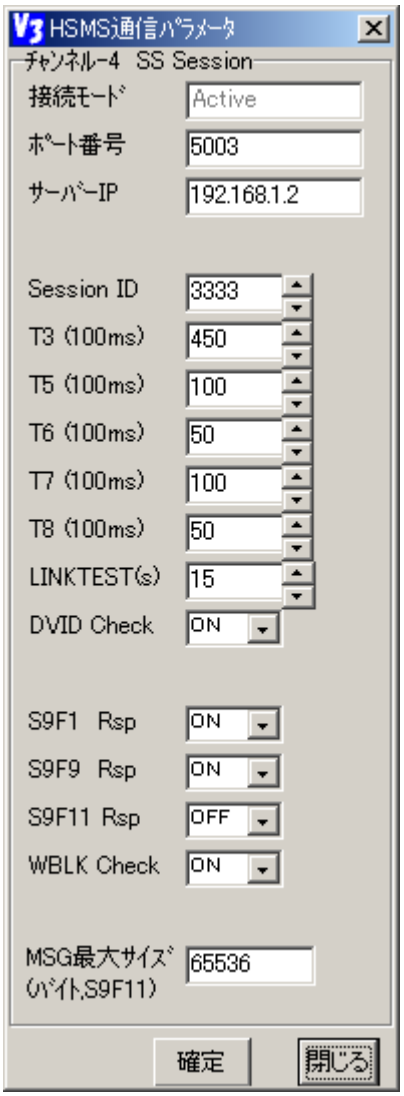
MSG 最大ｻｲｽﾞは、S9F11RSP=ON で受信ﾒｯｾｰｼﾞのｻﾞｲｽﾞが MSG 最大ｻｲｽﾞを超えた場合に、S9F11 を送信します。

（3）HSMS-SSプロトコル

各エンティティ（接続モード）の設定画面は以下のようになります。







① CH-3 の REMOTE ACTIVE の意味は、PASSIVE ﾁｬﾝﾈﾙが接続できるﾘﾓｰﾄの ACTIVE ﾎﾟｰﾄの情報です。

この画面の例では、IP=192.168.1.56 のﾘﾓｰﾄが、CH-2 への接続ができることを意味しています。

（注）通信環境定義ﾌｧｲﾙで PASSIVE CH の定義の中で、CLIENT=ANY の指定があれば、PASSIVE ﾎﾟｰﾄは、IP の制限なくﾘﾓｰﾄからの接続を受け付けます。

② LINKTEST の間隔時間の単位は秒(SEC)です。

③ S9Fx、ﾌﾞﾛｯｸﾁｪｯｸなどの設定は SECS の場合と同様です。

④ 接続ﾓｰﾄﾞの設定変更はこの画面ではできません。

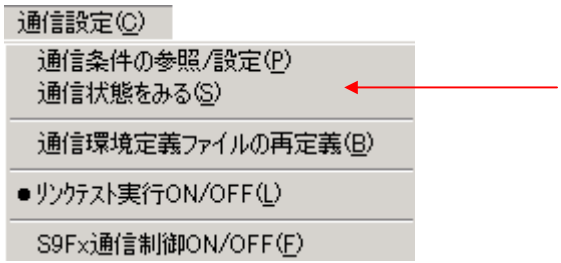
⑤ HSMS-GS の場合、最後のパラメータとして切断メッセージの指定があります。

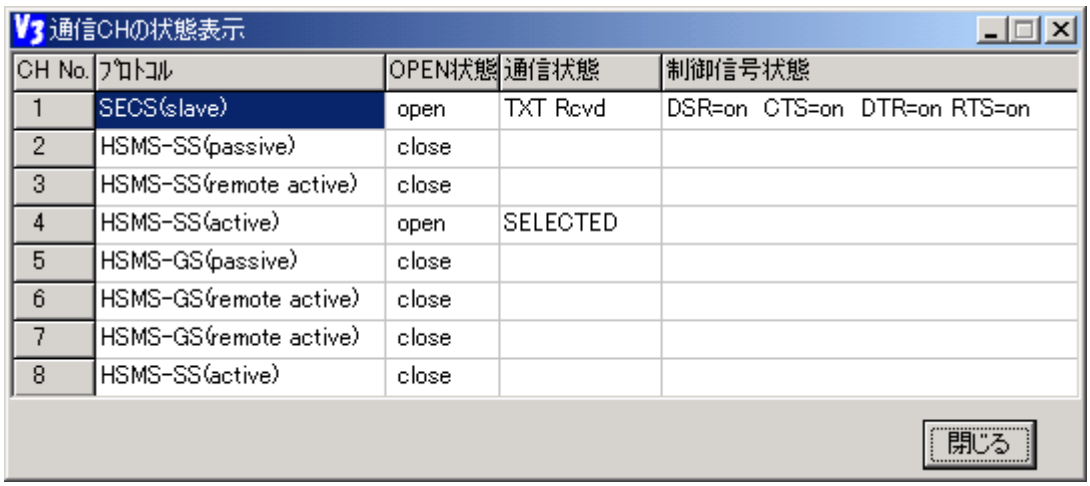
SEPARATE.req または DESELECT.req が選択できます。

４．６　チャンネル通信状態表示

全チャンネルの通信状態を画面で表示します。

操作は、メイン画面の**通信設定画面(C)**メニューの**通信状態をみる(S)**タブのクリックで表示できます。





通信状態欄には、プロトコルによって、

SECS

IDLE, ENQ_RCVD, LNE RCV, TXT RCVD, ENQ Wait(Multi-Blk), ENQ Wait(Retry)
ENQ_Sent, LEN_Sent, TXT_Sent, ACK_Rcvd が表示されます。

HSMS

IDLE, CONNECT_Wait, SELECT_Wait, SELECTED, DESELECT.Rsp_Wait が表示されます。

制御信号状態欄には、SECS(RS232C)の制御信号が表示されます。

DTR, DSR, CTS, RTS 信号の On/Off

４．７　通信プロトコルテスト

通信プロトコルテストは、プロジェクト実行中にそのプロジェクトが使用している通信チャンネルに対して、そのチャンネルに定められているＳＥＣＳまたはＨＳＭＳのプロトコルのテストを行います。

テストの目的は基本的に相手装置に対してプロトコル上のエラーの要因となる状態をシミュレータ側から作り出して、その要因に対して相手装置がプロトコル上、正しい制御を行っているかどうかを調べることにあります。

（１）プロトコルテストの開始

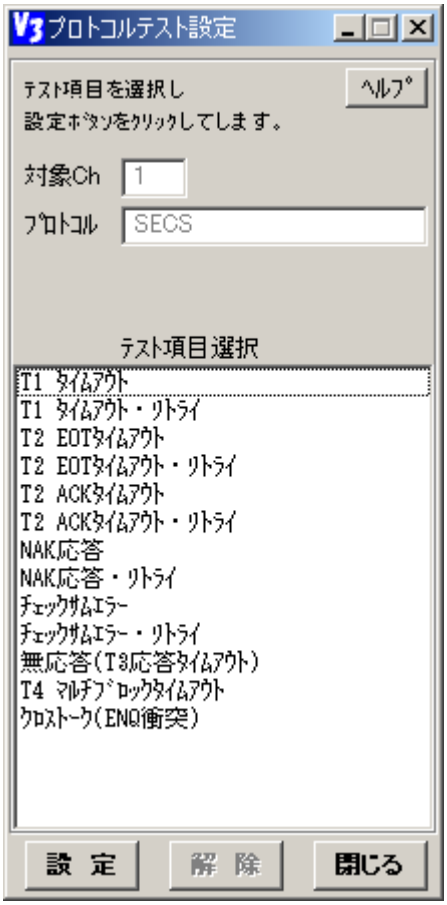
実行パネルのﾃｽﾄﾒﾆｭｰ(T)の中のﾌﾟﾛﾄｺﾙﾃｽﾄ(P)タブのクリックによって開始します。

開始すると、プロジェクトが使用しているチャンネルのプロトコルにしたがってテスト画面が表示されます。

（２）ＳＥＣＳプロトコルテストの設定

①　次の画面が表示されます。

ここで、テスト項目選択リストの中の１つの項目を選択し、ＯＫボタンをクリックすることによってテスト条件を設定することができます。



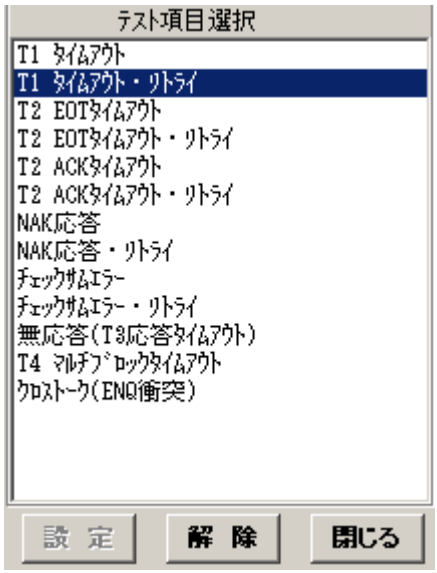
左の項目ﾘｽﾄの中で、”・ﾘﾄﾗｲ”が付いている、いないで、次のような違いがあります。

1. ﾘﾄﾗｲ無しの場合
   ﾘﾄﾗｲできるものは、ﾘﾄﾗｲ回数までｴﾗｰを起こします。最終的には、相手がﾘﾄﾗｲ　ｵｰﾊﾞｰで通信が終了することになります。
2. ﾘﾄﾗｲ有りの場合
   ｴﾗｰは１回だけ発生させ、その後、ﾘﾄﾗｲ時には正常に通信を行います。
   この場合、相手はﾘﾄﾗｲによって正常に通信が終了することになります。

② 設定が行われると、処理履歴画面に設定されたテスト項目が，例えばT1ﾀｲﾑｱｳﾄ･ﾘﾄﾗｲの場合、次のように表示されます。

```
08.10 10:00:27 ---- SECSﾌﾟﾛﾄｺﾙﾃｽﾄ T1 ﾀｲﾑｱｳﾄ・ﾘﾄﾗｲ 設定
```

更に、ボタンの状態は次のように、解除ボタンが有効になります。



③ 設定された項目のテストがそのタイミングになり、実施されると、その旨が表示されます。

```
08.10 10:06:49 ---- 通信ﾌﾟﾛﾄｺﾙﾃｽﾄ : ｴﾗｰ解除
08.10 10:06:49 CH-1 SECS 通信ﾃｽﾄ 送信 T1 ﾀｲﾑｱｳﾄ・ﾘﾄﾗｲﾃｽﾄ実施 st=10
```

実施とともに、設定は自動的に解除されます。そして、設定ボタンが自動的に有効になります。

④ テストをキャンセルしたい場合、テスト実施前に 解除ボタンをクリックすれば、テストを解除できます。

⑤ 下の表に、各テスト項目の内容を一覧表でまとめてありますので参照してください。

⑥ **閉じる**ボタンがクリックされると、設定されているテスト項目は解除されて本画面を終了します。

ＳＥＣＳテスト項目一覧表

番号	テスト項目名	ｼﾐｭﾚｰﾀ　　相手装置	テストの内容
1.	T1 ﾀｲﾑｱｳﾄ	ENQ ---> <--- EOT LEN --->（T1 ﾀｲﾑｱｳﾄ） <--- NAK 以下、このｼｰｹﾝｽをﾘﾄﾗｲ回数分続ける。	相手装置が T1 ﾀｲﾑｱｳﾄを検出し、NAK 送信をするかどうかを確認する。 ENQ, EOT のやり取り、ﾒｯｾｰｼﾞ長を送信した後、ﾃｷｽﾄを送信しないで、NAK を待つ。
2.	T1 ﾀｲﾑｱｳﾄ・ﾘﾄﾗｲ	ENQ ---> <--- EOT LEN --->（T1 ﾀｲﾑｱｳﾄ） <--- NAK ENQ ---> <--- EOT LEN --->TXT --->CKSM ---> <--- ACK	相手装置が T1 ﾀｲﾑｱｳﾄを検出し、NAK 送信をするかどうかを確認する。 また、その後に正常に送信されるﾒｯｾｰｼﾞを受信できるかどうかを確認する。
3.	T2 EOT ﾀｲﾑｱｳﾄ	<--- ENQ 無応答 （T2 ﾀｲﾑｱｳﾄ） 以下、このｼｰｹﾝｽをﾘﾄﾗｲ回数分続ける。	相手装置からの ENQ に対し、EOT を応答しないで、相手装置が T2 ﾀｲﾑｱｳﾄの検出を行い、その後、所定回数のﾘﾄﾗｲを行うかどうか、また、ﾘﾄﾗｲｵｰﾊﾞｰで送信を断念するかどうかを確認する。
4.	T2 EOT ﾀｲﾑｱｳﾄ・ﾘﾄﾗｲ	<--- ENQ 無応答 （T2 ﾀｲﾑｱｳﾄ） <--- ENQ EOT --->LEN <--- TXT <--- CKSM ACK	相手装置からの ENQ に対し、EOT を応答しないで、相手装置が T2 ﾀｲﾑｱｳﾄの検出を行い、その後、ﾘﾄﾗｲで正常にﾒｯｾｰｼﾞの送信を完結できるかどうかを確認する。
5.	T2 ACK ﾀｲﾑｱｳﾄ	<--- ENQ EOT --->LEN <--- TXT <--- CKSM 無応答 （T2 ﾀｲﾑｱｳﾄ） 以下、このｼｰｹﾝｽをﾘﾄﾗｲ回数分続ける。	相手装置からの送信に対し、CKSM を受信した後，ACK を応答しないで、相手装置が T2 ﾀｲﾑｱｳﾄの検出を行い、その後、所定回数のﾘﾄﾗｲを行うかどうか、また、ﾘﾄﾗｲｵｰﾊﾞｰで送信を断念するかどうかを確認する。

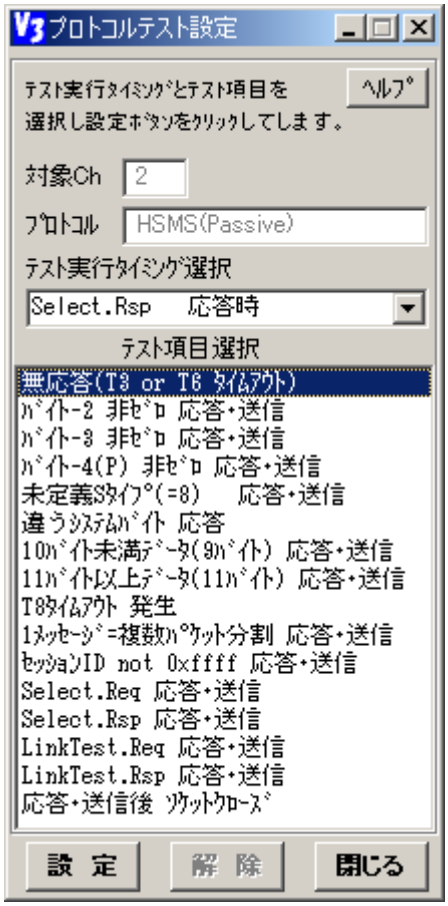
6.	T2 ACK ﾀｲﾑｱｳﾄ ・ﾘﾄﾗｲ	<---- ENQ EOT ----> <---- LEN <---- TXT <---- CKSM 無応答 (T2 ﾀｲﾑｱｳﾄ)	相手装置からの送信に対し、CKSM を受信した後、ACK を応答しないで、相手装置が T2 ﾀｲﾑｱｳﾄの検出を行い、その後、ﾘﾄﾗｲで正常にﾒｯｾｰｼﾞの送信を完結できるかどうかを確認する。
7.	NAK 応答	<---- ENQ EOT ----> <---- LEN <---- TXT <---- CKSM NAK ----> 以下、このｼｰｹﾝｽをﾘﾄﾗｲ回数分続ける。	相手装置からの送信に対し、CKSM を受信した後，NAK を応答したｹｰｽで、相手装置がその後、所定回数のﾘﾄﾗｲを行うかどうか、また、ﾘﾄﾗｲｵｰﾊﾞｰで送信を断念するかどうかを確認する
8.	NAK 応答・ﾘﾄﾗｲ	<---- ENQ EOT ----> <---- LEN <---- TXT <---- CKSM NAK ----> <---- ENQ EOT ----> <---- LEN <---- TXT <---- CKSM ACK ---->	相手装置からの送信に対し、CKSM を受信した後，NAK を応答したｹｰｽで、相手装置がその後、ﾘﾄﾗｲで正常にﾒｯｾｰｼﾞの送信を完結できるかどうかを確認する。
9.	ﾁｪｯｸｻﾑｴﾗｰ	ENQ ----> <---- EOT LEN ----> TXT ----> CKSM ----> (正しくない CKSM) <---- NAK 以下、このｼｰｹﾝｽをﾘﾄﾗｲ回数分続ける。	相手装置に正しくないﾁｪｯｸｻﾑﾃﾞｰﾀを送信した場合、相手装置が NAK を応答するかどうかを確認する。 また、引き続きﾘﾄﾗｲ回数分これを続けた場合の動作確認をする。
10.	ﾁｪｯｸｻﾑｴﾗｰ ・ﾘﾄﾗｲ	ENQ ----> <---- EOT LEN ----> TXT ----> CKSM ----> (正しくない CKSM) <---- NAK ENQ ----> <---- EOT LEN ---->	相手装置に正しくないﾁｪｯｸｻﾑﾃﾞｰﾀを送信した場合、相手装置が NAK を応答するかどうかを確認する。 また、引き続き正しいﾁｪｯｸｻﾑﾃﾞｰﾀを送った場合、相手装置が正しく受信できることを確認する。

		TXT ---> CKSM ---> (正しい CKSM) <--- ACK	
11.	無応答 (T3 ﾀｲﾑｱｳﾄ)	<--- 2 次 MSG 期待 の 1 次 MSG 無応答 (T3 ﾀｲﾑｱｳﾄ) T3 ﾀｲﾑｱｳﾄ 検出	相手装置からの２次ﾒｯｾｰｼﾞを期待する(W-bit=1)の１次ﾒｯｾｰｼﾞを受信しても、応答ﾒｯｾｰｼﾞを返さない場合、相手装置がT3 ﾀｲﾑｱｳﾄを検出することを確認する。
12.	T4 ﾏﾙﾁﾌﾞﾛｯｸ ﾀｲﾑｱｳﾄ	ﾏﾙﾁﾌﾞﾛｯｸの MSG BLK-1 ---> BLK-2 を送信しない。 (T4 ﾀｲﾑｱｳﾄ) T4 ﾀｲﾑｱｳﾄ 検出	ﾏﾙﾁﾌﾞﾛｯｸﾒｯｾｰｼﾞ送信時、第１ﾌﾞﾛｯｸのﾒｯｾｰｼﾞを送信した後、第２ﾌﾞﾛｯｸを送信しないｹｰｽで、相手装置が T4 ﾀｲﾑｱｳﾄ時間以上経過したことを検出するかどうかを確認する。
13.	ｸﾛｽﾄｰｸ (ENQ 衝突)	(1) ｼﾐｭﾚｰﾀ = Slave 相手装置 = Master ENQ --->| <--- ENQ EOT --->| <--- LEN . . ACK ENQ --->| <--- EOT . . <--- ACK (2) シミュレータ = Master 相手装置= Slave <--- ENQ ENQ --->| <--- EOT .	送信がｼﾐｭﾚｰﾀ、相手装置の間で、同時に行われようとしたｹｰｽ、即ち ENQ が衝突した ｹｰｽで、Slave 側が送信を一旦取りやめ、Master 側からのﾒｯｾｰｼﾞの受信に入り、受信が終了した後、Slave 側は、先に取りやめた送信を行うことを確認する。

		. <--- ACK <--- ENQ EOT ---> . . ACK --->	

（3）ＨＳＭＳ　Ｐａｓｓｉｖｅチャンネルのプロトコルテストの設定

① 次の画面が表示されます。ＨＳＭＳプロトコルテストは、プロトコルのタイミングによって、
対象となるテスト項目が違ってきます。



② まず、ﾃｽﾄ実行ﾀｲﾐﾝｸﾞを選ぶ必要があります。
選択肢として次の５つのタイミングがあり、この中から選択します。

ﾃｽﾄﾀｲﾐﾝｸﾞは文字通りのﾀｲﾐﾝｸﾞを意味します。

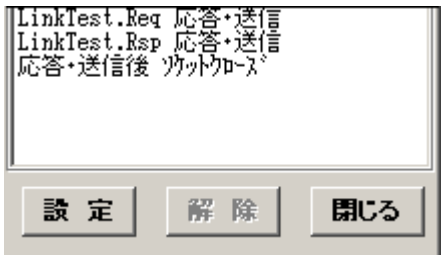
例えば、Select.Rsp を選択した場合、①のテスト項目選択画面が表示されます。

③ 後は、ＳＥＣＳプロトコルの場合と同様に、テスト項目を選択した上で、設定ボタンをクリックします。

④ 設定の後、処理履歴画面に設定された旨の表示が次のようにされます。

```
08.10 13:50:51 ---- HSMS ﾌﾟﾛﾄｺﾙﾃｽﾄ Select.Rsp　応答時:ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答･送信 設定
```

設定の後は、設定ボタンが無効になり、解除ボタンが有効になります。設定できるテスト項目は同時には１個だけです。



⑤　設定されたテストが実施されると次のように処理履歴画面が表示されます。

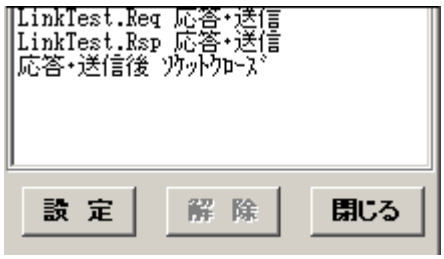
08.10 14:24:32 ---- HSMS ﾌﾟﾛﾄｺﾙﾃｽﾄ Select.Rsp　応答時:ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答･送信 ﾃｽﾄ実施

ＨＳＭＳプロトコルテストでは、設定されたテスト項目は、テストが実施されても自動的には解除されません。⑥の**解除**ボタンで解除されるまで**設定**は有効ということになります。

⑥　設定の解除は、**解除**ボタンのクリックで行います。

08.10 14:31:08 ---- 通信ﾌﾟﾛﾄｺﾙﾃｽﾄ : ｴﾗｰ解除

設定ボタンが再び有効になります。



⑦　次の表に各テスト実行タイミングとテスト項目についてまとめてあります。

ＨＳＭＳ―Passive テスト実行タイミングとテスト項目一覧表

（注）ＨＳＭＳ－ＳＳでは、Deselect.Req,Deselect.Rsp は適用されません。
ＨＳＭＳ－ＧＳでは、Passive または Remote-Active ポートに適用されます。

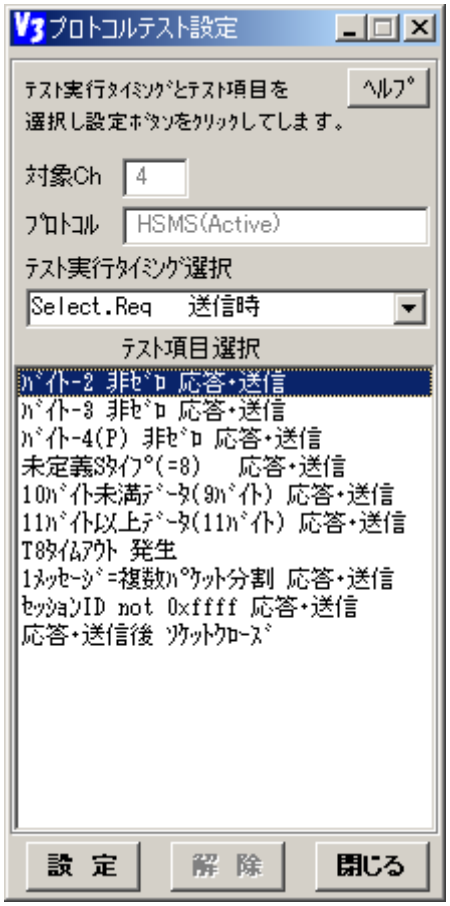
（○が付いている組合せのテストが準備されています。）

番号	ﾃｽﾄ項目名	Select.Rsp 応答時	Linktest.Req 送信時	Linktest.Rsp 応答時	ﾃｷｽﾄﾃﾞｰﾀ 送信時	ﾃｷｽﾄﾃﾞｰﾀ 送信時	Deselect.Req 送信時	Deselect.Rsp 応答時	注　釈
1.	無応答(T3 or T6 ﾀｲﾑｱｳﾄ)	○		○		○	○	○	
2.	ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答/送信	○	○	○			○	○	
3.	ﾊﾞｲﾄ-3 非ｾﾞﾛ 応答/送信	○	○	○			○	○	
4.	ﾊﾞｲﾄ-4(P)非ｾﾞﾛ 応答/送信	○	○	○	○	○	○	○	
5.	未定義 S(=8) 応答・送信	○	○	○	○	○	○	○	
6.	違うｼｽﾃﾑﾊﾞｲﾄ 応答	○	○	○		○	○	○	
7.	10 ﾊﾞｲﾄ未満ﾃﾞｰﾀ (9 ﾊﾞｲﾄ)応答･送信	○	○	○	○		○	○	
8.	11 ﾊﾞｲﾄ以上ﾃﾞｰﾀ (11 ﾊﾞｲﾄ)応答・送信	○	○	○			○	○	
9.	T8 ﾀｲﾑｱｳﾄ発生	○	○	○	○	○	○	○	
10.	1 ﾒｯｾｰｼﾞ=複数ﾊﾟｹｯﾄ 分割応答・送信	○	○	○	○	○	○	○	
11.	1 ﾊﾟｹｯﾄ複数ﾒｯｾｰｼﾞ 応答・送信				○	○			
12.	ｾｯｼｮﾝ ID not 0xffff 応答・送信	○	○	○					
13.	Select.Req	○		○		○		○	

	応答・送信								
14.	Select.Rsp 応答・送信	○		○		○		○	
15.	Linktest.Req 応答・送信	○		○		○		○	
16.	Linktest.Rsp 応答・送信	○		○		○		○	
17.	64KB 長ﾃﾞｰﾀﾒｯｾｰｼﾞ 応答・送信				○	○			
18.	応答・送信後 即時ｿｹｯﾄｸﾛｰｽﾞ	○	○	○	○	○	○	○	
19.	Deselect.Req 応答・送信	○		○		○		○	
20.	Deselect.Rsp 応答・送信	○		○		○		○	

（4）Ａｃｔｉｖｅチャンネルのプロトコルテストの設定

① 次の画面が表示されます。ＨＳＭＳプロトコルテストは、プロトコルのタイミングによって、対象となるテスト項目が違ってきます。



② まず、テスト実行タイミング を選ぶ必要があります。
次の５つのタイミング選択肢があり、この中から選択します。

ﾃｽﾄﾀｲﾐﾝｸﾞは文字通りのﾀｲﾐﾝｸﾞを意味します。

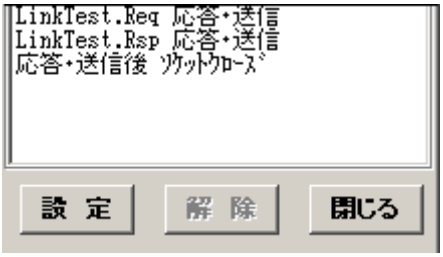
例えば、Select.Req 送信時 を選択した場合、①のテスト項目選択画面が表示されます。

③ 後は、ＳＥＣＳのプロトコルの場合と同様に、テスト項目を選択した上で、設定ボタンをクリックします。

④ 設定の後、処理履歴画面に設定された旨の表示が次のようにされます。

```
08.10 13:50:51 ---- HSMS ﾌﾟﾛﾄｺﾙﾃｽﾄ Select.Rsp   応答時:ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答・送信 設定
```

設定の後は、設定ボタンが無効になり、解除ボタンが有効になります。設定できるテスト項目は同時には１個だけです。



⑤　設定されたテストが実施されると次のように処理履歴画面が表示されます。

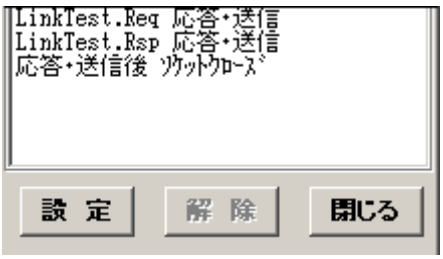
```
08.10 14:24:32 ---- HSMS ﾌﾟﾛﾄｺﾙﾃｽﾄ Select.Rsp    応答時:ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答･送信 ﾃｽﾄ実施
```

ＨＳＭＳプロトコルテストでは、設定されたテスト項目は、テストが実施されても自動的には解除されません。⑥の解除ボタンで解除されるまで設定は有効ということになります。

⑥　設定の解除は、解除ボタンのクリックで行います。

```
08.10 14:31:08 ---- 通信ﾌﾟﾛﾄｺﾙﾃｽﾄ : ｴﾗｰ解除
```

設定ボタンが再び有効になります。



⑦　次ページの表に各テスト実行タイミングとテスト項目について説明します。

HSMS　Active - テスト実行タイミングとテスト項目一覧表

(注)　ＨＳＭＳ－ＳＳでは、Deselect.Req,Deselect.Rsp は適用されません。

(○が付いている組合せのテストが準備されている。)

番号	ﾃｽﾄ項目名	Select.Req 送信時	Linktest.Req 送信時	Linktest.Rsp 応答時	ﾃｷｽﾄﾃﾞｰﾀ 送信時	ﾃｷｽﾄﾃﾞｰﾀ 送信時	Deselect.Req 送信時	Deselect.Rsp 応答時	注　釈
1.	無応答(T3 or T6 ﾀｲﾑｱｳﾄ)			○		○		○	
2.	ﾊﾞｲﾄ-2 非ｾﾞﾛ 応答/送信	○	○	○			○	○	
3.	ﾊﾞｲﾄ-3 非ｾﾞﾛ 応答/送信	○	○	○			○	○	
4.	ﾊﾞｲﾄ-4(P)非ｾﾞﾛ 応答/送信	○	○	○	○	○	○	○	
5.	未定義S(=8) 応答・送信	○	○	○	○	○	○	○	
6.	違うｼｽﾃﾑﾊﾞｲﾄ 応答		○	○		○		○	
7.	10ﾊﾞｲﾄ未満ﾃﾞｰﾀ (9ﾊﾞｲﾄ)応答･送信	○	○	○	○		○	○	
8.	11 ﾊﾞｲﾄ以上ﾃﾞｰﾀ(11 ﾊﾞｲﾄ)応答・送信	○	○	○			○	○	
9.	T8 ﾀｲﾑｱｳﾄ発生	○	○	○	○	○	○	○	
10.	1ﾒｯｾｰｼﾞ=複数ﾊﾟｹｯﾄ 分割応答・送信	○	○	○	○	○	○	○	
11.	1 ﾊﾟｹｯﾄ複数ﾒｯｾｰｼﾞ応答・送信				○	○			
12.	ｾｯｼｮﾝ ID not 0xffff 応答・送信	○	○	○			○	○	HSMS-SS の場合
13.	Select.Req 応答・送信			○		○		○	

14.	Select.Rsp 応答・送信			○		○		○	
15.	Linktest.Req 応答・送信			○		○		○	
16.	Linktest.Rsp 応答・送信			○		○		○	
17.	64KB 長ﾃﾞｰﾀﾒｯｾｰｼﾞ応答・送信				○	○			
18.	応答・送信後 即時ｿｹｯﾄｸﾛｰｽﾞ	○	○	○	○	○	○	○	
19.	Deselect.Req 応答・送信			○		○		○	
20.	Deselect.Rsp 応答・送信			○		○		○	

４．８　連続送信テスト

相手装置に対する連続送信テストを行うための機能の操作について説明します。

実行パネルの操作で説明した１次メッセージの送信は、オペレータがメッセージを指定した後、送信ボタンをクリックしなければ送信できませんでしたが、この連続通信テスト機能を使用すれば、複数の１次メッセージをオペレータが一々介在することなく一定周期で連続的に相手装置に送信することができます。

また、連続送信テスト中に相手装置から送られてくるメッセージに対する応答は、自動応答モードにしておけば、自動的に行われます。
（但し、応答メッセージになりうるメッセージが２個以上定義されている場合、オペレータに応答メッセージの選択と送信を求めてきます。）

（1）連続送信テスト画面の表示

① 実行パネル画面のﾃｽﾄ機能(C)の中の連続送信ﾃｽﾄ(S)メニューのクリックで連続送信テスト画面が表示されます。

（2）送信メッセージの指定方法

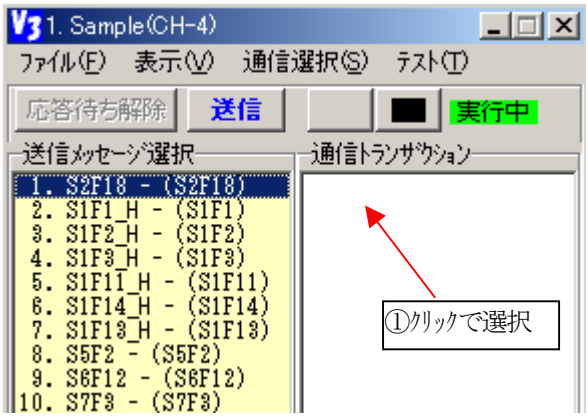
送信対象にするメッセージならびに周期などの設定方法には、２つの方法があります。それは、1.画面からの設定の方法、2.送信メッセージなどの情報が予め保存されている送信リストファイルから読出す方法です。
送信リストファイルの拡張子は（.TST）に決められています。ファイルの内部表現については後述します。

① 画面上で設定する方法

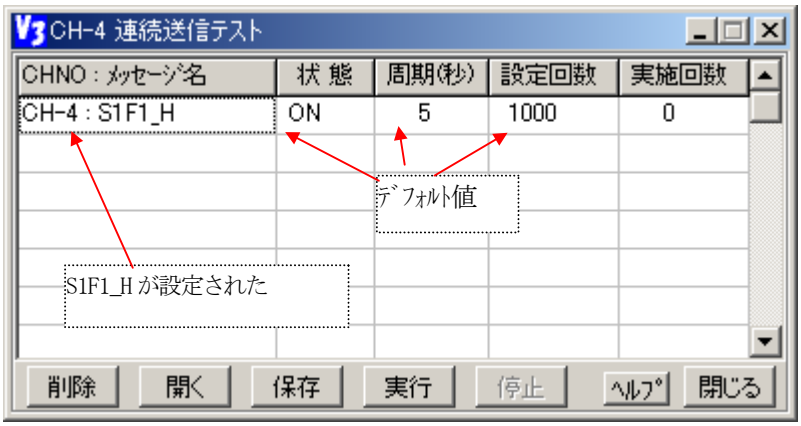
マニュアル通信画面の送信ﾒｯｾｰｼﾞを参照しながらマウスを使って操作します。
例えば、S1F1_H メッセージを送信メッセージとして設定したい場合、次の順に操作を行います。

a. 左側の送信 MSG(選択)画面上の 1. S1F1_H を、送信するときと同じようにクリックし、選択します。
1. S1F1_H メッセージ名が反転表示になります。

b. この後、右側の連続送信画面のメッセージ名の欄で、空白のセルをダブルクリックします。
これで、S1F1_H の選択ができ、連続送信テスト画面のクリックしたセルに S1F1_1 が表示され、同時に状態、周期、設定回数欄にはデフォルト値が表示されます。

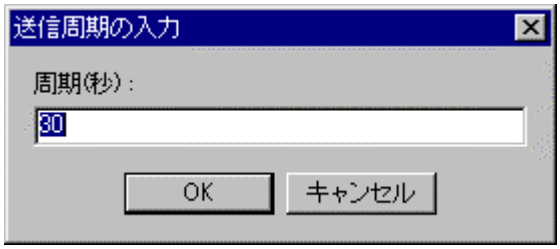


S1F1_H の選択後の画面は次のようになります。



c. 状態 ON/OFF の変更は、ON(OFF)のセル位置をダブルクリックするだけで、状態が切り替わります。
OFF の状態では、テストを実行してもそのメッセージは送信されません。

d. 周期(秒)、設定回数の値の変更は、やはり、そのセル位置をダブルクリックし、ポップアップされた次の入力画面上で設定値を変更し、OK で変更します。



e. 画面上のテスト設定情報の保存
送信メッセージの設定が終わったあと、このテスト情報を送信リストファイルに保存することができます。
保存は 保存 ボタンのクリックの後、通常のファイル保存ダイアログ画面操作で保存します。
このとき、ファイル名には自動的に拡張子".TST"が付けられます。

② 送信リストファイルによる送信情報の設定
①-e で保存された、あるいは、テキストエディターで予め作成されている送信リストファイルを開いて設定します。
画面の**開く**ボタンをクリックし、通常のファイルを開くダイアログ画面の操作で開きます。開いた結果、③-b の画面のように、送信メッセージ名リストが表示されます。

（3）送信メッセージの送信対象からの除外

設定した送信メッセージを送信対象から外したい場合は、状態を OFF にするか、または、外したいメッセージ名を削除します。

削除する場合は、削除したいメッセージ名をクリックで選択し、その後、削除ボタンをクリックします。
そこで、確認画面がポップアップされますので、はい(YES)ボタンをクリックして削除します。

（4）連続送信テストの実行

連続送信テストは実行ボタンのクリックで開始されます。
このとき、画面の実施回数の欄の値はすべて=0 にリセットされます。
実際に送信されるメッセージは、状態が ON になっており、設定回数が 0 でないものです。実行が開始されると、送信された回数が実施回数欄に表示され、メッセージごとに設定されている周期で送信が繰り返されます。そして、設定回数に達するか、そのメッセージの状態が OFF にされるか、または停止ボタンがクリックされるまで続けられます。但し、HSMS 通信の場合、実際には、相手装置との接続が確立した後に送信が行われます。送信されたメッセージはオペレータによって送信された場合と同様に通信履歴、レポートファイルへの記録、T3 タイマー監視が行われます。

（5）連続テストの停止

停止ボタンのクリックで送信テストを停止します。

（6）送信リストファイルのフォーマット

送信リストファイルの拡張子は ″.TST″です。
ファイルはテキストファイルであり、テキストエディターで作成することができます。また、画面で設定した内容を、一旦ファイルに保存し、後でそれを再度開いて使用することもできます。

ファイルには、１メッセージの指定を１行で、次のフォーマットで表現します。

<MSG 名>,<ON/OFF>,<周期>,<ﾃｽﾄ回数>

<MSG 名> ： メッセージ定義ファイルで定義されているメッセージ名を指定します。
<ON/OFF> ： 読込んだときの送信可能状態の指定を行います。
ON で、実行時に送信される。
OFF で、実行時に送信されない。(ON に変更されるまで)
<周期> ： 送信周期（間隔）を秒単位で指定します。
<ﾃｽﾄ回数> ： テストしたい回数を指定します。
それぞれの項目の間は、’,’（カンマ）で区切ります。

例 S1F5_H,ON,25,1000 は、S1F5_H メッセージを 25 秒周期で 1000 回送信し、読み込み時は、送信可能状態であることを意味します。

４．９　マニュアルシーケンステスト

予め、通信シナリオの順に送信メッセージと送信条件情報を並べてファイルに登録しておき、後で、その登録したファイルを開いてメッセージを順次自動送信するための機能と操作について説明します。

シーケンステスト機能を使用することによって、同じシナリオのテストを何度も繰り返し行う場合、いちいちメッセージを捜す手間がなくなり、効率よく簡単にテストすることができます。

本機能は以前のマニュアルシーケンステスト機能に代わるものであり、機能をさらに進化させ、シーケンス内のメッセージを送信するタイミング条件を決める受信メッセージ情報を設定することができます。もちろん、以前に作成したシーケンスファイルも使用できます。

受信メッセージ情報は、以下のとおりです。

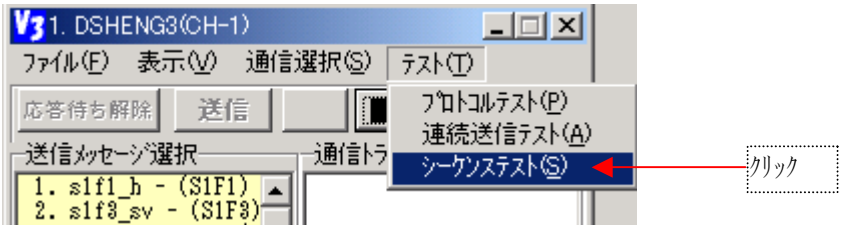
① 相手からの特定メッセージ ID SxFy(例 S6F11)
② □のメッセージが有するデータアイテム名（メッセージ定義ファイルで付けられた名前, 例 S6F11 の”CEID”）
③ □のデータアイテムの値（例 CEID の値 1200）
④ データアイテムは最大２個まで設定することができます。

シーケンスを画面上で設定した後、**連続実行**ボタンをクリックすることによって、シーケンス順に指定された条件が成立したら、指定された遅延時間をとりながら連続的にメッセージを送信させることができます。

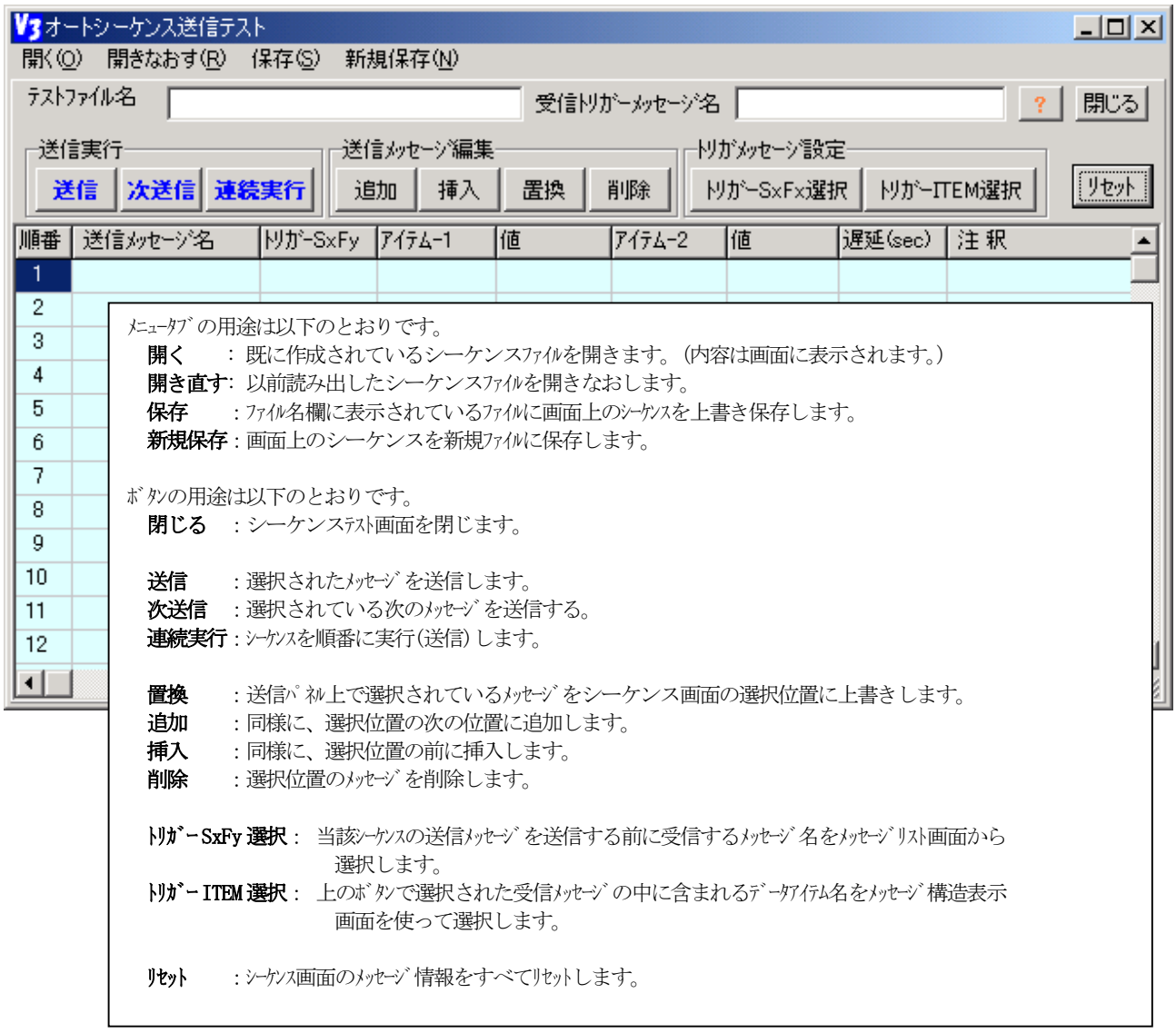
本機能の操作の前提としては、プロジェクトが開かれていることです。

（1）シーケンステスト画面の表示

□　実行パネル画面のﾃｽﾄ(T) メニューの中の**ｼｰｹﾝｽﾃｽﾄ(S)** タブのクリックで始めます。



次のシーケンス画面が表示されます。



シーケンス内容の設定は上のボタンを使いながら、例えば、次のような画面に設定します。

順番	送信ﾒｯｾｰｼﾞ名	ﾄﾘｶﾞｰSxFy	ｱｲﾃﾑ-1	値	ｱｲﾃﾑ-2	値	遅延(sec)	注 釈
1	s3f17_ContentMap	S6F11	ceid	1200			5	LoadAssociated
2	s3f17_Proceed	S6F11	ceid	1201			5	LoadWaitingForHost

グリッド画面の各セルには以下の情報を設定します。

上の画面のｶﾗﾑを左側から説明します。
順番： 送信するﾒｯｾｰｼﾞの順番号で固定です。番号順に実行されます。
送信ﾒｯｾｰｼﾞ名： 送信したいﾒｯｾｰｼﾞ名(送信ﾊﾟﾈﾙ画面操作から選択します。)
ﾄﾘｶﾞｰSxFy： 上の送信ﾒｯｾｰｼﾞ送信の条件になる受信ﾒｯｾｰｼﾞの ID(SxFy)
ｱｲﾃﾑ-1： ﾄﾘｶﾞｰSxFy ﾒｯｾｰｼﾞに含まれるﾃﾞｰﾀｱｲﾃﾑの名前
値： ｱｲﾃﾑ-1 の値を指定します。
ｱｲﾃﾑ-2： ﾄﾘｶﾞｰSxFy ﾒｯｾｰｼﾞに含まれるﾃﾞｰﾀｱｲﾃﾑの名前
値： ｱｲﾃﾑ-2 の値を指定します
遅延(sec)： 送信の条件が揃った後ここで指定した時間の遅延を取った後に送信します。
注釈： 操作用ﾒﾓです。

シーケンスは**送信ﾒｯｾｰｼﾞ名**カラムが空白のところで終わりになります。

連続実行において**送信ﾒｯｾｰｼﾞ名**で指定されたメッセージの送信条件は次のように判断されることになります。

① ﾄﾘｶﾞｰSxFy のカラムに受信条件がない場合
　送信遅延カラムの指定時間後に送信します。

② ﾄﾘｶﾞｰSxFy のカラムに受信条件がある場合で、ｱｲﾃﾑ-1, 2 のデータアイテム名がない場合
　SxFy のメッセージを受信後、遅延カラムの指定時間後に送信します。

③ ﾄﾘｶﾞｰSxFy のカラムに受信条件があり、ｱｲﾃﾑ-1, 2 のデータアイテム名のどちらかまたは両方ある場合
　SxFy のメッセージで指定データアイテムの値が一致した後、遅延カラムの指定時間後に送信します。

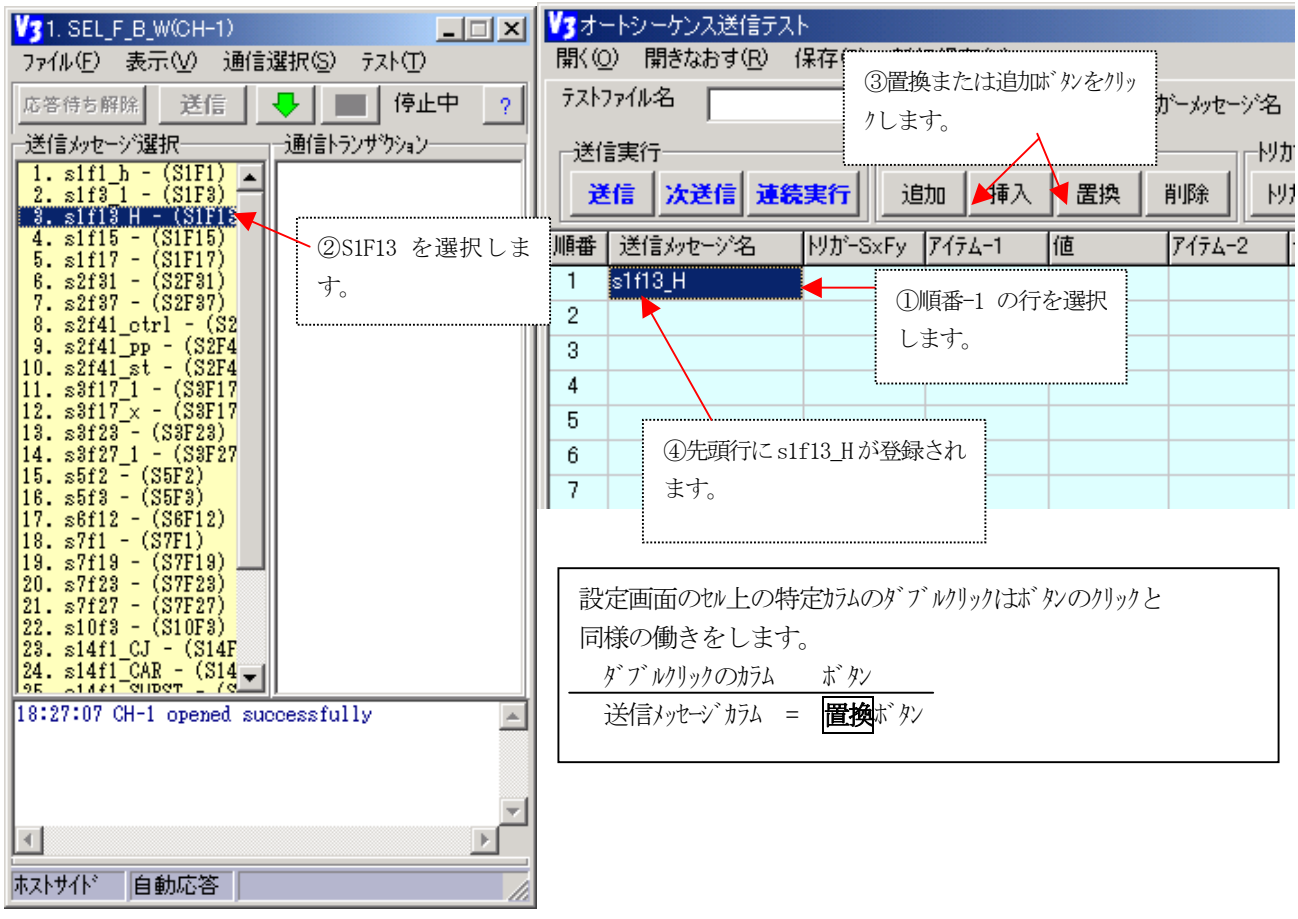
（2）シーケンスの設定方法

（2）－１送信メッセージの登録方法

送信パネル画面との連携で登録することができます。即ち、送信パネル上のメッセージ名を選択しておき、シーケンス画面の追加、挿入、置換ボタンをクリックすることでシーケンス画面上にメッセージを登録することができます。

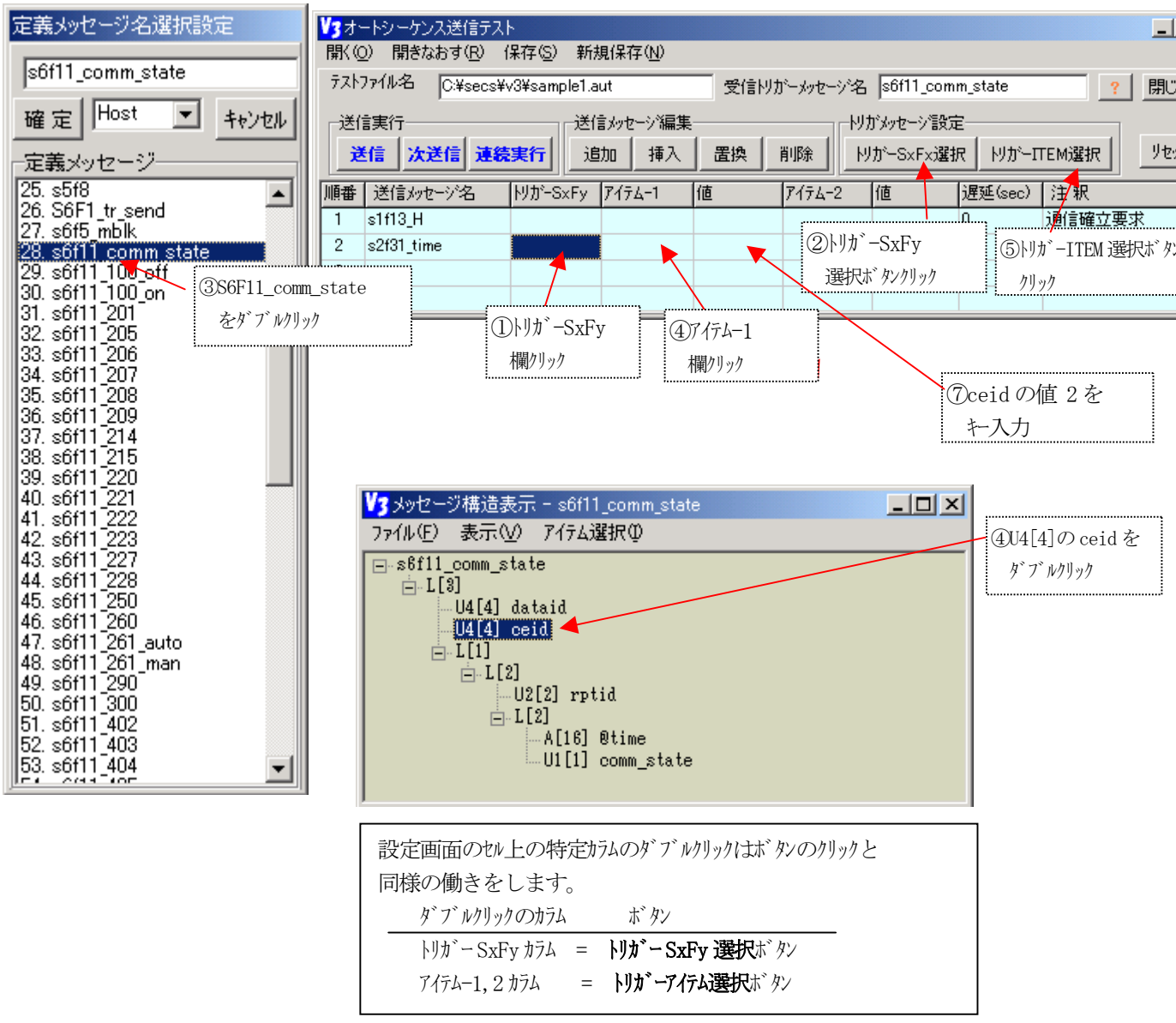
登録するメッセージは、基本的に１次メッセージだけです。（２次メッセージは、自動生成プログラムが１次メッセージを受信した後自動的に応答を返してくれます。応答メッセージとして複数の候補がある場合は、応答メッセージを選択するための画面がポップアップされ、選択操作で応答することができます。）

例えばS1F13_H をシーケンスの先頭に登録するためには、次の〇内の番号順に操作します。



（2）－２送信トリガー条件の設定

ここでは s2f31_time メッセージ送信条件として S6F11 CEID=2 のメッセージの受信を設定することにします。〇の番号順に操作をしてください。



設定画面のｾﾙ上の特定ｶﾗﾑのﾀﾞﾌﾞﾙｸﾘｯｸはﾎﾞﾀﾝのｸﾘｯｸと同様の働きをします。

ﾀﾞﾌﾞﾙｸﾘｯｸのｶﾗﾑ		ﾎﾞﾀﾝ
ﾄﾘｶﾞｰSxFy ｶﾗﾑ	=	**ﾄﾘｶﾞｰSxFy 選択**ﾎﾞﾀﾝ
ｱｲﾃﾑ-1, 2 ｶﾗﾑ	=	**ﾄﾘｶﾞｰｱｲﾃﾑ選択**ﾎﾞﾀﾝ

設定結果はつぎの画面のようになります。

なお、送信メッセージ名を始めとする各設定項目は、名セルに直接キー入力することににによってメッセージ名を設定することができます。ただし、入力ミスを避けるため、送信パネル画面や選択画面との連携での設定をお勧めします。

以下、シナリオ（通信シーケンス）の順に送信メッセージを登録していきます。
また、実際に、シーケンス実行中に、各登録メッセージに対する注釈（メモ）を加えることができます。この注釈は、メッセージ名の右側の注釈欄にキー入力によって直接設定してください。

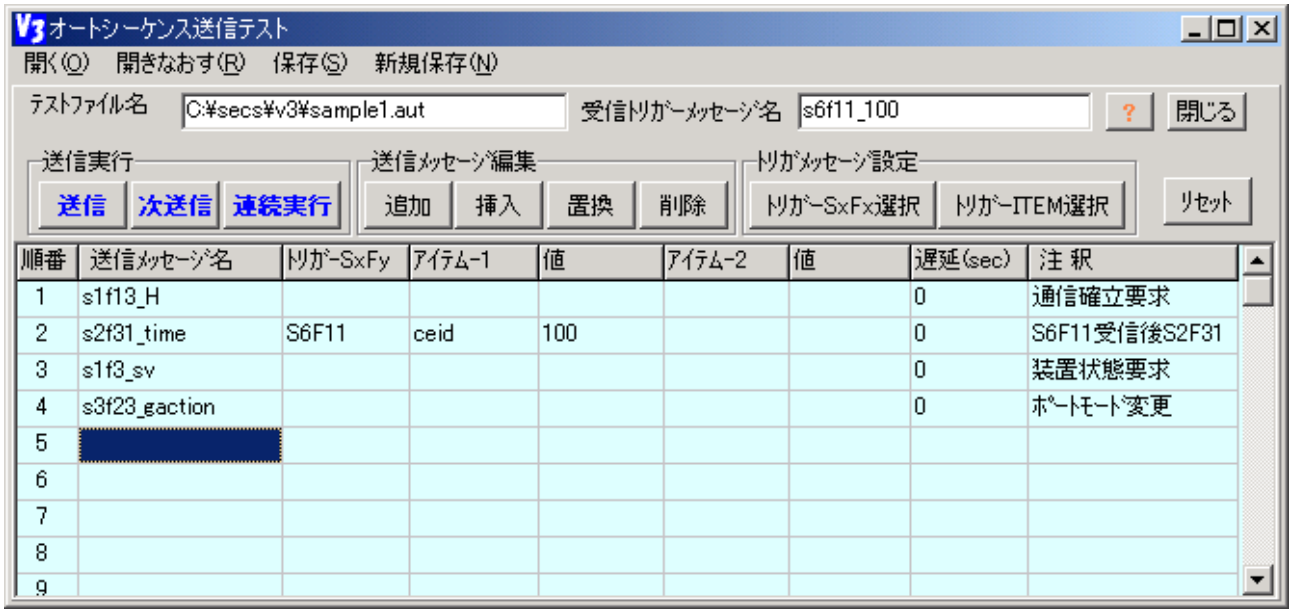
それから、遅延時間(sec)も設定できます。この遅延時間は，連続ボタンによる送信時、当該シーケンスのメッセージを送信する前に、指定された遅延時間を取ってから送信します。(遅延時間は、当該ステップ送信前に、相手から時間がかかる場合などのための待機時間です、 0秒でも構いません。)

（2）－3　設定結果

例えば、次のようなシナリオに対するホスト側のシーケンスの設定は、次のようになります。

ホスト側	ﾄﾗﾝｻﾞｸｼｮﾝ	装置側	ｼｰｹﾝｽ設定対象（○）
1. S1F13H 送信 Establish Comm Request	S1F13 ---> <== S1F14		○
2. S6F11 受信 CEID=2(ｺﾝﾄﾛｰﾙ状態ｲﾍﾞﾝﾄ)受信	<-- S6F11 S6F12 ==>	Event Report Send	(自動応答) S2F31 送信ﾄﾘｶﾞｰ
3. S2F31 送信 Date and Time Set Request 送信	S2F31 --> <== S2F32		○
4. S1F3_1 送信 Selected Equipment Satatus Request	S1F3 --> <== S1F4		○
5. S3F23 送信 Change Access	S3F23 --> <== S2F24		○

表の中のシーケンス設定対象○のもの(ホストが送信する１次メッセージ)を順に登録します。注釈も入れると例えば、次のような画面になります。



（3）登録シーケンスをファイルに保存します。

後で再利用できるようにするため、作成した画面の内容をファイルに保存しておきます。
最初は、**新規保存**ﾒﾆｭｰﾀﾌﾞのクリックで保存します。
後で、開きなおし、内容を変更し、同じファイルに保存する場合には、**保存**ﾒﾆｭｰﾀﾌﾞを使ってください。

（4）シーケンスファイルのロード

（3）で保存したシーケンスファイルを再びロードするためにメニューの開くまたは開き直すタブのクリックで行います。

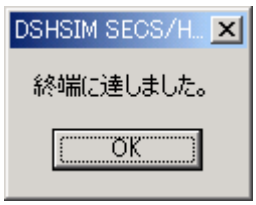
（5）シーケンス実行操作

送信は、相手装置とのプロトコルが成立している状態（HSMS では Selected 状態）でしか有効になりませんので、ボタンが有効になったらクリックできます。

（5）－１　**連続実行**ボタンクリック

① 最初、シーケンスの先頭の行を選択します。
② 連続実行ボタンをクリックします。その後、シーケンスの順に、トリガー受信メッセージがあればそれを待機し、遅延時間をとりながらメッセージを送信します。
③ もし、途中で送信を中止したい場合は、連続実行ボタンをもう一度クリックしてください。

連続操作実行中は、連続実行ボタンが点滅状態になります。
最後のメッセージに達すると、ビープ音３回鳴らして知らせてくれます。



（5）－２　**送信**、**次送信**ボタンのクリックによる操作

個別にメッセージを送信するための操作です。
（この場合、遅延時間とは関係なく即時送信します。）

① はじめに、シーケンスの最初の行を選択します。
② そして、送信タイミングがきたら、送信ボタンをクリックし、メッセージを送信します。

③　以下、次送信ボタンをタイミングを計りながら、順次クリックし、送信します。
③の操作を繰り返します。

④　最後のメッセージを送信し終わった後、次送信ボタンをクリックすると、終わったことを画面で教えてくれます。

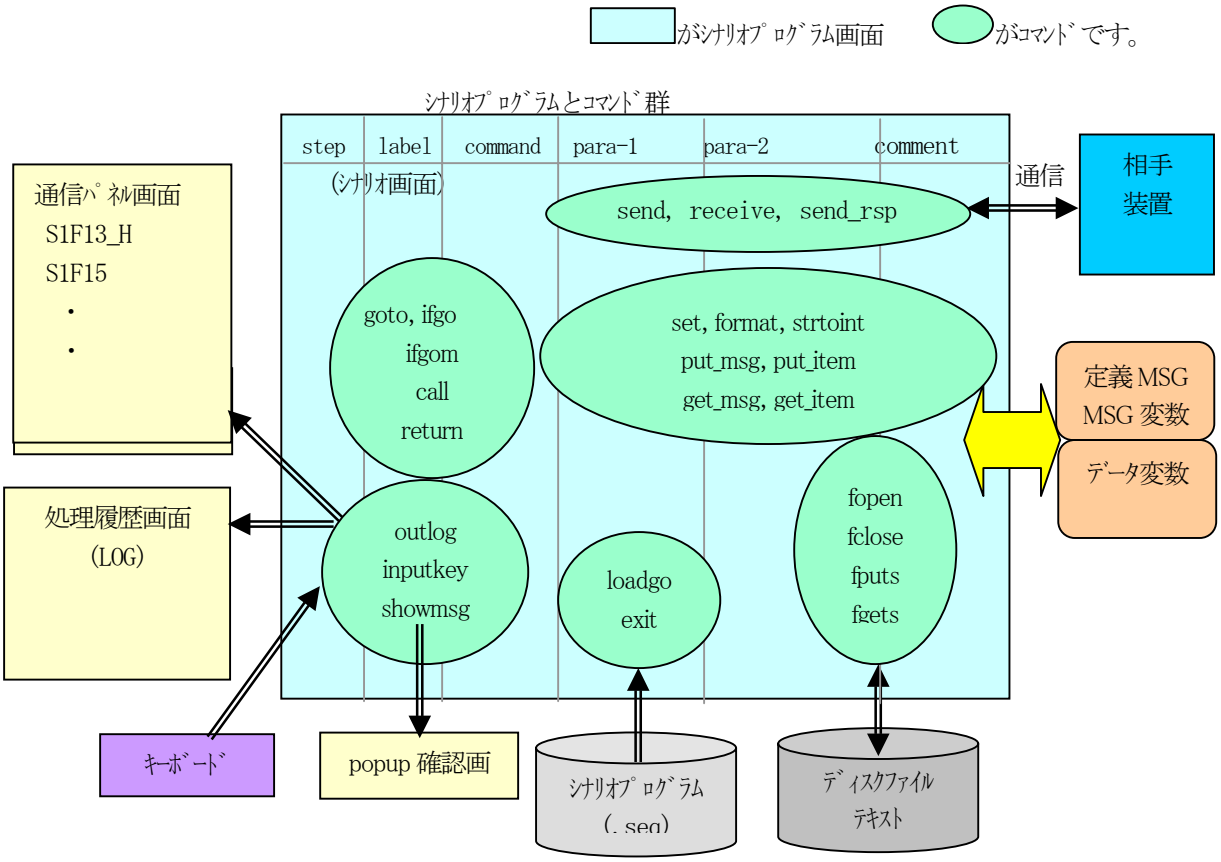
４．１０　シナリオプログラムによるテスト

（1）はじめに

従来のシーケンステストを新たに強化した機能がシナリオテスト機能です。
本機能はユーザがシナリオ言語を使って簡単に画面上にプログラムを組み込み通信シナリオのテストを実現することです。

シナリオ言語コマンドで作成されたプログラムを“**シナリオプログラム**”と呼びます。シナリオプログラムファイルの拡張子を “**. seq**” で表します。

シナリオプログラム、コマンドとその周辺との関連を表すと大体以下の図のようになります。



［特長］

・なによりも手短にプログラムを組んでテストできます。
・画面の表の中に、シナリオコマンド（26 種類）を使って簡単にプログラミングし実行することができます。
・通信パネルが持っている情報との連携操作でプログラム編集ができ、コマンドを記憶しなくても組めます。
　プログラム内ではデータ変数、定義メッセージ、メッセージ変数などを参照することになります。
　コマンド、変数などはポップアップ一覧表画面からマウスクリックで選択することができます。
・メッセージ編集、データ編集機能ができます。
　put_item, get_item, format, strtoint コマンドなどを使用します。
・ディスクファイルの文字列データを read/write することができます。
　ファイルを使ってより多くの情報をセットしてテストでき、結果もファイルに記録することができます。
・コンパイルや生成操作無しで実行開始できます。
・実行中はどのステップのコマンドが実行されているかを画面で追跡することができます。
・勿論、作ったシナリオプログラムは、ファイルに保存し、後で再ロードして実行することができます。

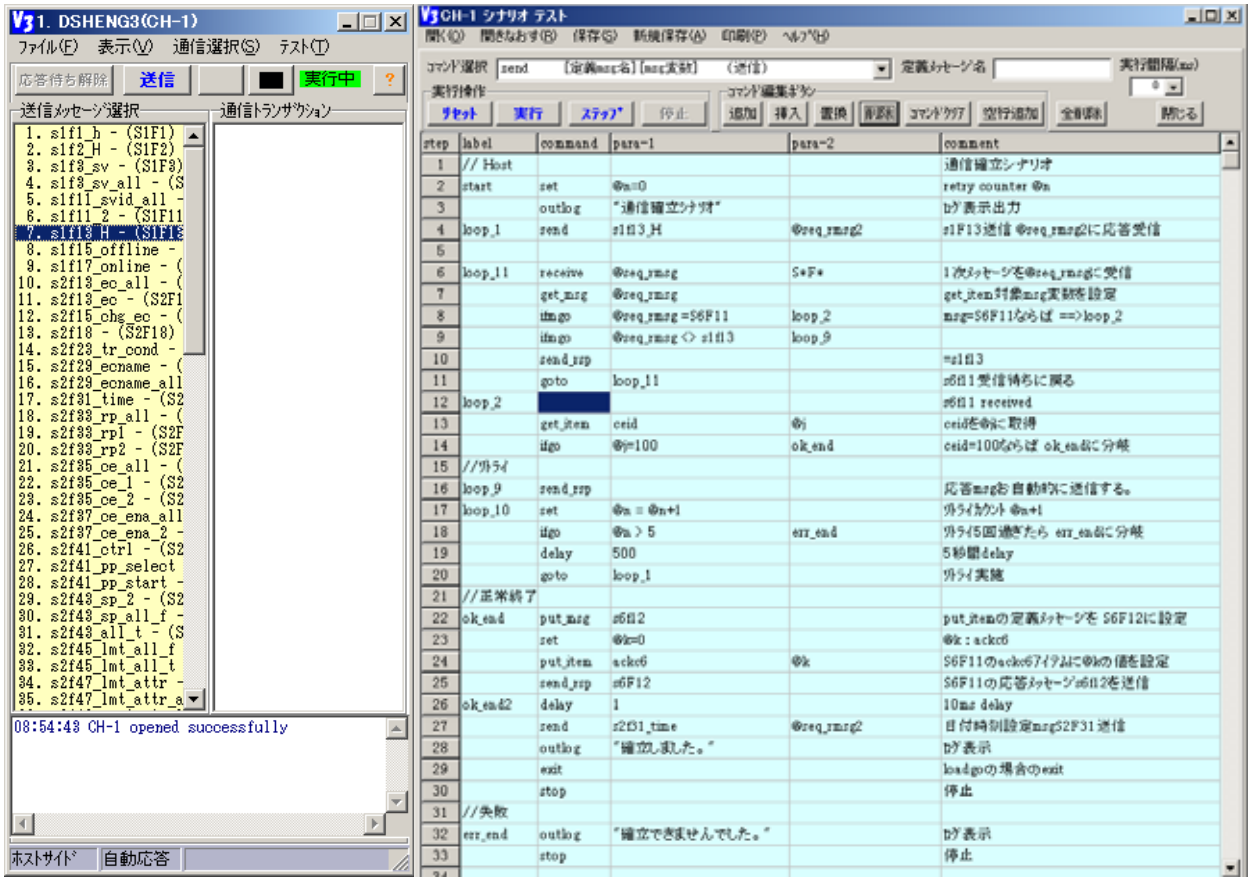
・loadgo ｺﾏﾝﾄﾞを使って他のシナリオプログラムをサブルーチンのようにロードし実行させることができます。
　exit コマンドで元の呼出したプログラムに戻ります。
　複数のシナリオを順番に呼出ながら実行できます。

簡単なシナリオとテスト画面例を挙げます。以下、例として使用するメッセージ名は dsheng3.msg ファイルに定義されているものを使用しています。

ホスト側が、最初に実行する GEM の通信確立と立ち上がりシーケンスのシナリオを次のように行うと仮定します。

ホスト	通信	装置	コメント
S1F13	---------->		S1F13 を送信
	<==========	S1F14	
	<----------	S6F11	ceid=100 の S6F11 を受信
S6F12	==========>		
S2F31	---------->		時刻設定
	<==========	S2F32	

このシナリオプログラムは例えば、次のようになります。この後、画面の**実行**ボタンのクリックでこの画面上でプログラムをすぐに実行することができます。



（2）シナリオ言語コマンド形式とコマンド一覧表

コマンド形式は上で画面で示したように次の６つのフイールドから成ります。

step	label	command	para-1	para-2	comment

	ﾌｲｰﾙﾄﾞ	説明
1	step	プログラムの順番を示すステップ番号です。
2	label	分岐コマンド(goto, call など)の飛び先になるラベルです。英数字と'_'からなる文字列です。 頭に **“//”**（ｽﾗｯｼｭ文字が 2 個）が付いている行のステップはコメントとみなされ実行されません。
3	command	コマンド名のフイールドです。 ブランクの場合は、なにもしないで、スキップされます。
4	para-1	コマンドに付属する１番目のパラメータです。コマンドによっては必要ない場合もあります。
5	para-2	コマンドに付属する２番目のパラメータです。コマンドによっては必要ない場合もあります。
6	comment	コメント（注釈）のためのフイールドです。コマンドの実行には影響しません。

なお、大文字と小文字の区別ですが、

1. ラベル、コマンド、データ変数名、メッセージ変数名などでの英数字の大文字小文字の区別はしません。
2. 書式編集コマンドの<format>を指定する ”“ 内の文字列の文字については大文字と小文字の区別をします。

コマンドには以表のとおり２６種類のコマンドがあります。

注）ﾊﾟﾗﾒｰﾀ, para-1, 2 欄の”なし”の記述はﾊﾟﾗﾒｰﾀがないことを意味します。

no.	command	para-1	para-2	ｺﾏﾝﾄﾞ機能
1	send	<def msg>	<msg 変数>	<def msg>で指定された定義ﾒｯｾｰｼﾞを送信します。 受信した応答ﾒｯｾｰｼﾞは<msg 変数>内に格納します。 送信ﾁｬﾝﾈﾙはﾌﾟﾛｼﾞｪｸﾄ(通信ﾊﾟﾈﾙの)で指定されたﾁｬﾝﾈﾙになります。 ｴﾗｰを検出した場合、その旨を表示し停止します。 送信前に、<def msg>内のﾃﾞｰﾀｱｲﾃﾑの値をput_msg, put_itemｺﾏﾝﾄﾞを使って設定変更することができます。 <msg 変数>のﾃﾞﾌｫﾙﾄの変数は seq_rmsg2 です。 例 send　S1F13_H　seq_rmsg2
2	receive	<msg 変数>	<SxFy>	1次ﾒｯｾｰｼﾞを受信し、それを<msg 変数>に格納します。 受信したいﾒｯｾｰｼﾞがある場合その ID を<SxFy>で指定します。 <SxFy>の指定が S*F*の場合は、全ての 1 次ﾒｯｾｰｼﾞが受信対象になります。 <msg 変数>のﾃﾞﾌｫﾙﾄの変数は seq_rmsg です。 例 receive seq_rmsg　S6F11

3	send_rsp	<def msg> (省略可)	なし	receive ｺﾏﾝﾄﾞで受信した1次ﾒｯｾｰｼﾞに対する2次ﾒｯｾｰｼﾞを応答送信します。 <def msg>に指定があるときは、そこで指定された定義ﾒｯｾｰｼﾞを送信します。 <def msg>が省略された場合は、ｼﾐｭﾚｰﾀが受信した1次ﾒｯｾｰｼﾞから自動的に応答ﾒｯｾｰｼﾞを選び出し、そのときの定義ﾒｯｾｰｼﾞの内容で送信します。 例 send_rsp S6F11
4	put_msg	<def msg>		put_item ｺﾏﾝﾄﾞでﾃﾞｰﾀｱｲﾃﾑの値を設定したい定義ﾒｯｾｰｼﾞ名として<def msg>を指定します。本ｺﾏﾝﾄﾞの後、put_item でﾃﾞｰﾀｱｲﾃﾑを設定し、send または send_rsp ｺﾏﾝﾄﾞを使って送信します。 例 put_msg S6F12
5	put_item	<item 名>	<ﾃﾞｰﾀ変数>	put_msg ｺﾏﾝﾄﾞで指定した定義ﾒｯｾｰｼﾞに含まれる<item 名>ﾃﾞｰﾀｱｲﾃﾑの値を<ﾃﾞｰﾀ変数>の値に設定します。 <item 名>のﾃﾞｰﾀｱｲﾃﾑﾀｲﾌﾟは次のものが対象です。 数値ﾀｲﾌﾟ:整数だけをｻﾎﾟｰﾄします。 B, BOOLEAN, S1,S2,S4,U1,U2,U4 です。 文字ﾀｲﾌﾟ: A, J <ﾃﾞｰﾀ変数>のﾃﾞｰﾀﾀｲﾌﾟは次の2種類です。 数値ﾀｲﾌﾟ : 整数だけをｻﾎﾟｰﾄします。 INT1,INT2,INT4,UINT1,UINT2,UINT4 文字ﾀｲﾌﾟ: CHAR, STRING 例 set @k=1 put_item ackc6 @k @kの値をackc6ｱｲﾃﾑに設定します。
6	get_msg	<msg 変数>		get_item ｺﾏﾝﾄﾞでﾃﾞｰﾀｱｲﾃﾑの値を取得したいﾒｯｾｰｼﾞ変数を<msg 変数>に指定します。受信したﾒｯｾｰｼﾞに含まれるﾃﾞｰﾀｱｲﾃﾑ値を取得するｺﾏﾝﾄﾞ get_item の前に使用します。 send または receive ｺﾏﾝﾄﾞで指定したﾒｯｾｰｼﾞ変数を設定してください。 例 get_msg @seq_rmsg
7	get_item	<item 名>	<ﾃﾞｰﾀ変数>	get_msg ｺﾏﾝﾄﾞで指定したﾒｯｾｰｼﾞ変数から<item 名>で指定されたﾃﾞｰﾀｱｲﾃﾑの値を<ﾃﾞｰﾀ変数>に取得するためのｺﾏﾝﾄﾞです。 (本ｺﾏﾝﾄﾞはﾌﾟﾛｸﾞﾗﾏｰが、get_msg で指定された変数内のﾒｯｾｰｼﾞに含まれるﾃﾞｰﾀｱｲﾃﾑを既知していることを前提にしています。) <item 名>のﾃﾞｰﾀｱｲﾃﾑﾀｲﾌﾟは次のものが対象です。 数値ﾀｲﾌﾟ:整数だけをｻﾎﾟｰﾄします。 B, BOOLEAN, S1,S2,S4,U1,U2,U4 です。 文字ﾀｲﾌﾟ: A, J <ﾃﾞｰﾀ変数>のﾃﾞｰﾀﾀｲﾌﾟ 数値ﾀｲﾌﾟ : 整数だけをｻﾎﾟｰﾄします。 INT1,INT2,INT4,UINT1,UINT2,UINT4 文字ﾀｲﾌﾟ: CHAR, STRING 例 get_item ceid @n2 ceid の値を@n2 に取得します。

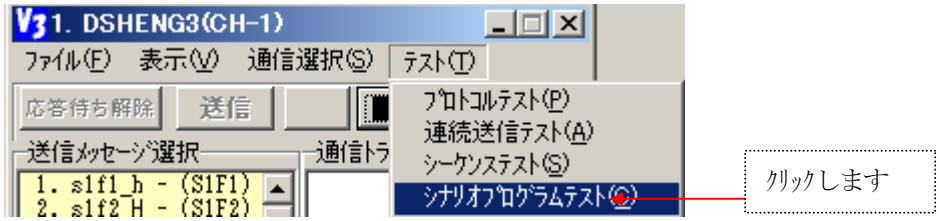
8	goto	<label>	なし	<label>で指定されたﾗﾍﾞﾙのｽﾃｯﾌﾟに無条件にｼﾞｬﾝﾌﾟします。 例 loop1 . goto loop1
9	ifgo	<変数> < <比較値> <= = >= > <>	<label>	<変数>の値と<比較値>とを比較して、比較演算子(<,<=, =,>=,>,<>)の条件が満たされた場合に<label>で指定されたﾗﾍﾞﾙのｽﾃｯﾌﾟにｼﾞｬﾝﾌﾟします。 比較対照は整数の変数についてのみ行います。(文字列の比較はできません。) <比較値>としてﾘﾃﾗﾙの値または整数変数を指定できます。 例 ifgo @i=1 la_equal ifgo @i<>@j la_notequal
10	ifmgo	<msg 変数> = <SxFy> <>	<label>	<msg 変数>に含まれるSECS-IIﾒｯｾｰｼﾞのﾒｯｾｰｼﾞIDの値が比較演算子(=,<>)の条件を満たした場合に<label>で指定されたﾗﾍﾞﾙのｽﾃｯﾌﾟにｼﾞｬﾝﾌﾟします。 例 ifmgo seq_rmsg1=S6F11 lab_s6f11
11	call	<label>	なし	<label>で指定されたﾗﾍﾞﾙから始まるｻﾌﾞﾙｰﾁﾝをｺｰﾙします。 ｻﾌﾞﾙｰﾁﾝからの戻りには return ｺﾏﾝﾄﾞを使います。 例 call read_file . read_file fgets @buff2 @buff3 . . return
12	return	なし	なし	call されたｻﾌﾞﾙｰﾁﾝから call したプログラムに戻るためのｺﾏﾝﾄﾞです。
13	delay	<遅延時間値>	なし	<遅延時間値>(10ms 単位)の時間だけ処理をしないで時間経過を待ちます。 <遅延時間値>はﾘﾃﾗﾙ(数値)または整数変数で指定します。
14	stop	なし	なし	ﾌﾟﾛｸﾞﾗﾑの実行を停止します。 (停止後、編集が可能になります。)
15	set	<変数>=<式>	なし	<式>で表された計算式を実行し、結果を<変数>で指定されたﾃﾞｰﾀ変数に格納します。 変数は整数または String ﾀｲﾌﾟの変数名を指定してください。 <式>は、整数演算式またはﾘﾃﾗﾙ(=値そのもの)で与えます。 例 set @i = @i+@j+15 set @buff3 = “This is a sample”

16	strtoint	〈変数〉	〈文字列〉	〈文字列〉を整数の数値に変換します。 〈文字列〉は、ﾘﾃﾗﾙまたは文字ﾀｲﾌﾟの変数を指定します。 例 @i = strtoint 100 @i = strtoint @buff3
17	format	〈文字列変数〉	〈編集書式文字列〉	〈編集書式文字列〉に指定されている書式に基づいて指定された変数ﾃﾞｰﾀを文字列に編集変換し、〈文字列変数〉で指定された変数に格納します。 〈編集書式文字列〉には、**二重引用符(")**で囲まれた編集書式と、ｶﾝﾏ(,)で区切られた 1 個以上の変数名を指定します。 例 format "@i=%d @buff3=%s" @i,@buff3 編集書式については、書式付編集についてを参照してください。
18	outlog	〈編集書式文字列〉	なし	〈編集書式文字列〉に指定されている書式に基づいて編集し、処理履歴画面ならびに送信ﾊﾟﾈﾙのﾒﾓに表示出力します。 〈編集書式文字列〉については 17 の format ｺﾏﾝﾄﾞと同じです。 なお、文字列だけの指定でもかまいません。 例 outlog "これから開始します。" 表示は "*** outlog：〈文字列〉 ***"のように文字列の前後に *** outlog と ***を付けて行います。
19	showmsg	〈編集書式文字列〉	なし	〈編集書式文字列〉に指定されている書式に基づいて編集し、その文字列をﾎﾟｯﾌﾟｱｯﾌﾟ画面に表示させることができます。ﾎﾟｯﾌﾟｱｯﾌﾟ画面は警告表示などに使用できます。 例 showmsg "通信確立ｼﾅﾘｵ実行に失敗しました。"
20	keyinput	〈文字列変数〉	〈表示文字列〉	ｷｰから文字列を入力し、〈文字列変数〉に格納します。〈表示文字列〉はｷｰ入力画面の題名になります。 例 keyinput @buff "ﾚｼﾋﾟ ID を入力してください。"
21	fopen	〈ﾌｧｲﾙ名〉	r または w (省略可)	〈ﾌｧｲﾙ名〉に与えられたﾌｧｲﾙをﾊﾟﾗﾒｰﾀｰ2 の指定ﾓｰﾄﾞで開きます。 〈ﾌｧｲﾙ名〉にはﾘﾃﾗﾙまたは文字列変数を使用することができます。ﾘﾃﾗﾙの場合は二重引用符(")で囲んでください。 para-2 の意味は、'r'が read, 'w'が write になります。para-2 が省略(ﾌﾞﾗﾝｸ)された場合には r とみなします。 ﾌｧｲﾙはﾃｷｽﾄﾌｧｲﾙだけを対象とします。(ﾊﾞｲﾅﾘﾌｧｲﾙには使用できません。) 22,23 の fgets, fputs ｺﾏﾝﾄﾞの前に必ず本ｺﾏﾝﾄﾞでﾌｧｲﾙを開いてください。 例 fopen "c:¥test¥rcp_data.txt" r

22	fgets	〈ファイル名〉	〈文字列変数〉	〈ファイル名〉のファイルから1行分の文字列を読んで〈文字列〉変数に格納します。 ファイル名がリテラルの場合は二重引用符(“)で囲んでください。 文字列に含まれる改行(LF)、復帰(CR)文字は取り除かれた上で変数に格納されます。ファイル終端に達した場合、〈文字列変数〉の文字列長=0 の文字列を格納します。 ファイルは予め fopen ｺﾏﾝﾄﾞによって read ﾓｰﾄﾞで開かれていなければなりません。 例 fgets “c:¥test¥rcp_data.txt” @buff3
23	fputs	〈ファイル名〉	〈文字列 or 変数〉	〈文字列 or 変数〉の 1 行分の文字列を〈ファイル名〉で指定されたファイルに書込みます。文字列の最後に改行(LF)を付けて書込みます。 ファイル名がリテラルの場合は二重引用符(“)で囲んでください。 リテラル文字列を書込む場合は、二重引用符(“)で囲んでください。 ファイルは予め fopen ｺﾏﾝﾄﾞによって write ﾓｰﾄﾞで開かれていなければなりません。 例 fputs “c:¥test¥rcp_data.txt” “RCP-0001”
24	fclose	〈ファイル名〉	なし	先に fopen で開かれた〈ファイル名〉で指定されたファイルを閉じます。 ファイル名がリテラルの場合は二重引用符(“)で囲んでください。
25	loadgo	〈ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑ名〉	〈label〉 (省略可)	〈ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑ名〉で指定されたｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙをﾛｰﾄﾞし、〈label〉で指定されたﾗﾍﾞﾙ位置から実行します。〈label〉が省略された場合は先頭のｽﾃｯﾌﾟから実行されます。 ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ単位のｻﾌﾞﾙｰﾁﾝ呼出しのような機能になります。 後述の exit ｺﾏﾝﾄﾞに達するとﾛｰﾄﾞ要求元のﾌﾟﾛｸﾞﾗﾑに戻ります。 例 loadgo C:¥test¥sub1.seq start (sub1.seq をﾛｰﾄﾞし、“start”ﾗﾍﾞﾙから実行を開始します。)
26	exit	なし	なし	loadgo ｺﾏﾝﾄﾞでﾛｰﾄﾞ実行開始されたｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑが本ｺﾏﾝﾄﾞ exit に達すると、呼出元のﾌﾟﾛｸﾞﾗﾑに戻り、loadgo ｺﾏﾝﾄﾞの次のｽﾃｯﾌﾟを実行します。 もし、loadgo ｺﾏﾝﾄﾞで呼び出されていない場合は、ﾌﾟﾛｸﾞﾗﾑの最後のｺﾏﾝﾄﾞとみなし、停止します。

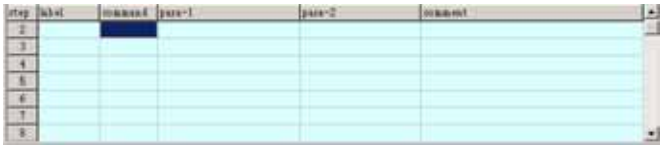
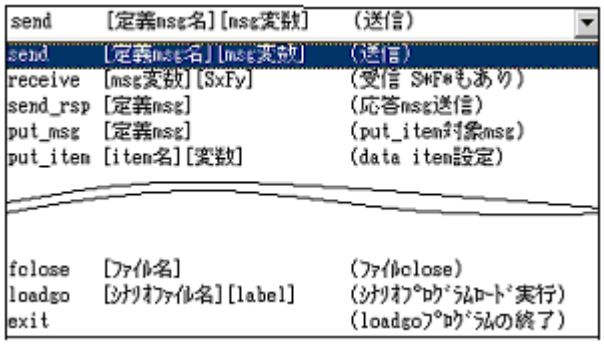
（3）シナリオテストの操作画面

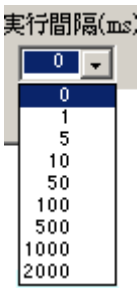
シナリオプログラムテストの画面は、実行パネルのテストメニューの ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾃｽﾄ(C) タブのクリックで開きます。



次のシナリオプログラム画面が表示されます。このとき、プロジェクト番号と使用する通信チャンネル番号が実行パネルから渡されます。

本画面で操作に関係するボタンなどの機能は次の通りです。

	ｸﾞﾙｰﾌﾟ	項目名	機能
1	ﾒﾆｭｰ	**開く(O)**	ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ(.seq)を開きます。
		開きなおす(R)	前に開いたｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ(.seq)を開きなおします。
		保存(S)	開いたｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑをﾌｧｲﾙに保存します。
		新規保存(A)	画面のｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑを別名のﾌｧｲﾙに保存します。
		印刷(P)	画面のｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑをﾌﾟﾘﾝﾄします。
		ﾍﾙﾌﾟ(H)	ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑについてのﾍﾙﾌﾟ画面を表示します。
2	表(grids)	**ﾌﾟﾛｸﾞﾗﾑｺﾏﾝﾄﾞ設定画面**	 各ｾﾙにｺﾏﾝﾄﾞ、ﾊﾟﾗﾒｰﾀを設定するための表です。 直接書込むことができます。 後述の3のｺﾝﾎﾞﾎﾞｯｸｽｺﾏﾝﾄﾞを選択し、追加、挿入ﾎﾞﾀﾝでｺﾏﾝﾄﾞを設定できます。 またﾊﾟﾗﾒｰﾀの設定には、ﾀﾞﾌﾞﾙｸﾘｯｸでﾎﾟｯﾌﾟｱｯﾌﾟﾊﾟﾗﾒｰﾀ設定画面を使うことができます。 削除などのﾎﾞﾀﾝでｺﾏﾝﾄﾞ行を編集することができます。
3	ｺﾝﾎﾞﾎﾞｯｸｽ	**ｺﾏﾝﾄﾞ選択**	設定したいｺﾏﾝﾄﾞを一覧表から選択します。 
4	編集ﾎﾞﾀﾝ	**追加**	ｺﾏﾝﾄﾞ選択のｺﾝﾎﾞﾎﾞｯｸｽのｺﾏﾝﾄﾞを現在位置の次の行に追加します。 追加されたｽﾃｯﾌﾟ以降のｺﾏﾝﾄﾞは後ろにｼﾌﾄされます。
		挿入	ｺﾏﾝﾄﾞ選択のｺﾝﾎﾞﾎﾞｯｸｽのｺﾏﾝﾄﾞを現在位置の前の行に挿入します。 挿入されたｽﾃｯﾌﾟ以降のｺﾏﾝﾄﾞは後ろにｼﾌﾄされます。
		置換	ｺﾏﾝﾄﾞ選択のｺﾝﾎﾞﾎﾞｯｸｽのｺﾏﾝﾄﾞを現在の行に設定します。 元のｺﾏﾝﾄﾞは消えます。
		削除	現在位置の行を削除します。 そして現在位置以降のｺﾏﾝﾄﾞを前の方にｼﾌﾄします。
		ｺﾏﾝﾄﾞｸﾘｱ	現在位置行のｺﾏﾝﾄﾞの表示をﾌﾞﾗﾝｸにします。 前後のｺﾏﾝﾄﾞは影響を受けません。
		空行追加	現在位置の次に空行を追加します。 追加されたｽﾃｯﾌﾟ以降のｺﾏﾝﾄﾞは後ろにｼﾌﾄされます。
		全削除	表の中のｺﾏﾝﾄﾞを全て削除します。 開かれていたｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙの名前の記憶も解消されます。
5	実行動作ﾎﾞﾀﾝ	**ﾘｾｯﾄ**	実行前にｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑの初期化処理を行うためのﾎﾞﾀﾝです。先に fopen ｺﾏﾝﾄﾞで開かれていたﾌｧｲﾙがあった場合は全て閉じます。また、実効開始ｽﾃｯﾌﾟ位置を１番目にします。継続ではない実行の場合は実行前に必ずｸﾘｯｸしてください。

		実行	画面に設定されているｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑを現在選択されている位置から実行開始します。実行中は実行されているｺﾏﾝﾄﾞのｾﾙが選択表示になります。（選択表示は色が反転します。） また、実行中は”開く”、”開きなおす”ﾒﾆｭｰと編集ﾎﾞﾀﾝが使用禁止状態になります。 実行は，stop ｺﾏﾝﾄﾞまたはその先に実行すべきｺﾏﾝﾄﾞが無くなったときまたは停止ﾎﾞﾀﾝがｸﾘｯｸされたとき、停止します。停止すると使用禁止されていたﾒﾆｭｰ、ﾎﾞﾀﾝが使用可能な状態に戻ります。
		ｽﾃｯﾌﾟ	現在位置のｺﾏﾝﾄﾞ１個だけを実行して停止します。実行したあと、現在位置は次のｽﾃｯﾌﾟに移ります。
		停止	実行中のﾌﾟﾛｸﾞﾗﾑを停止させます。 停止すると、実行中に使用禁止されていたﾒﾆｭｰ、ﾎﾞﾀﾝが使用可能な状態に戻ります。
6	定義ﾒｯｾｰｼﾞ名表示	**定義ﾒｯｾｰｼﾞ名**	put_item、get_item のために選択されている定義ﾒｯｾｰｼﾞ名が表示されます。これらｺﾏﾝﾄﾞのﾃﾞｰﾀｱｲﾃﾑの設定／取得対象ﾒｯｾｰｼﾞになります。
7	ｺﾏﾝﾄﾞ実行間隔時間選択	**実行間隔(ms)**	ｼﾅﾘｵﾌﾟﾛｸﾞﾗﾑ実行中にｺﾏﾝﾄﾞと次のｺﾏﾝﾄﾞの実行の間に時間的な遅延時間（単位ms）を作るための時間値選択のためのｺﾝﾎﾞﾎﾞｯｸｽです。 値=0(ms)が選択されていた場合は、遅延時間なしで連続的に実行します。遅延時間を長くすると、ﾌﾟﾛｸﾞﾗﾑの進行具合を目視しながら実行させることができます。 

（4）プログラミングの操作

新規にプログラムを作成するための操作を具体的に説明します。
プログラム例として、S5F1 メッセージの送信の作成について操作手順を記述します。
なお、定義メッセージとして S5F1_alarm を使用し、alid の値と altx データアイテムの内容をキー入力しながら S5F1 を送信し、受信した応答メッセージ S5F2 の ackc5 をログ表示することにします。

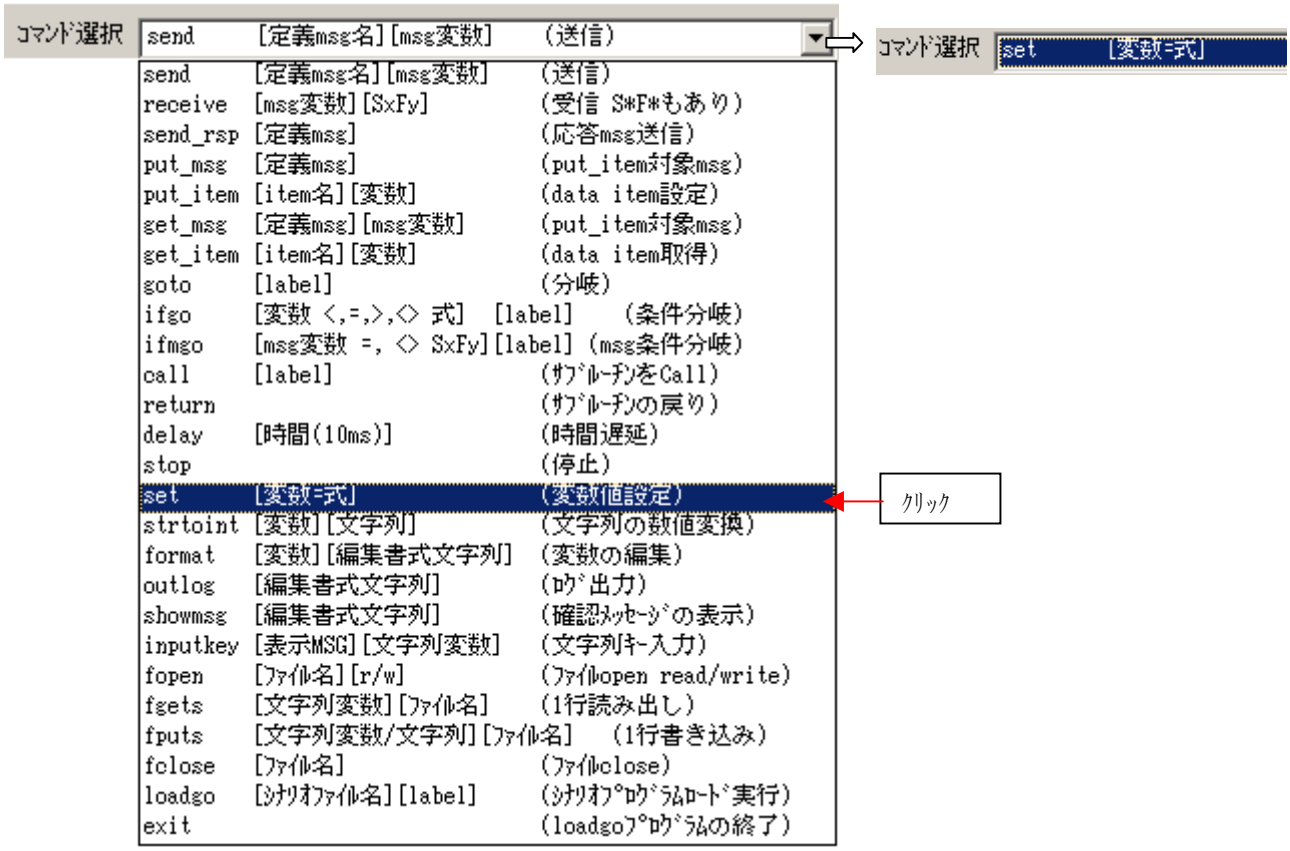
（4）-1 プログラムは画面上に以下のように組むことができます。

step	label	command	para-1	para-2	comment
1	//S5F1 test				S5F1 を 2 回送信して停止
2	start_s5f1				
3		set	@n=0		送信回数 @n=0
4	loop1	inputkey	“ALID を入力してください。	@buff1	ALID を入力
5		strtoint	@i2	@buff1	文字列を数値変換==>@i2
6		inputkey	“ALTX を入力してください。”	@buff2	ALTX を@buff2 に入力
7					
8		put_msg	s5f1_alarm		s5f1_alarm を put_item の対象にする
9		put_item	alid	@n	@i2 の値をﾃﾞｰﾀｱｲﾃﾑ alid に設定
10		put_item	altx	@buff2	@buff2 の値をﾃﾞｰﾀｱｲﾃﾑ altx に設定
11					
12		send	s5f1_alarm	@seq_rmsg2	s5f1_alarm を送信し、応答==>@seq_rmsg2
13		get_msg	@seq_msg2		ﾃﾞｰﾀｱｲﾃﾑ値取得 msg を@seq_msg2 にする
14		get_item	ackc5	@i3	ackc5 のｱｲﾃﾑ値を@i3 に取得する
15		outlog	“ackc5=%d”,@i3		ackc5 の値を表示する。
16					
17		set	@n=@n+1		
18		ifgo	@n < 2	loop1	@n<2 ならば繰り返し(loop1 へ)
19		stop			@n>=2 ならば停止（終了）
20					

（4）-2 上のプログラムを画面に設定する方法の一つは、表の中に順番に全コマンド行を書込んでいく方法です。

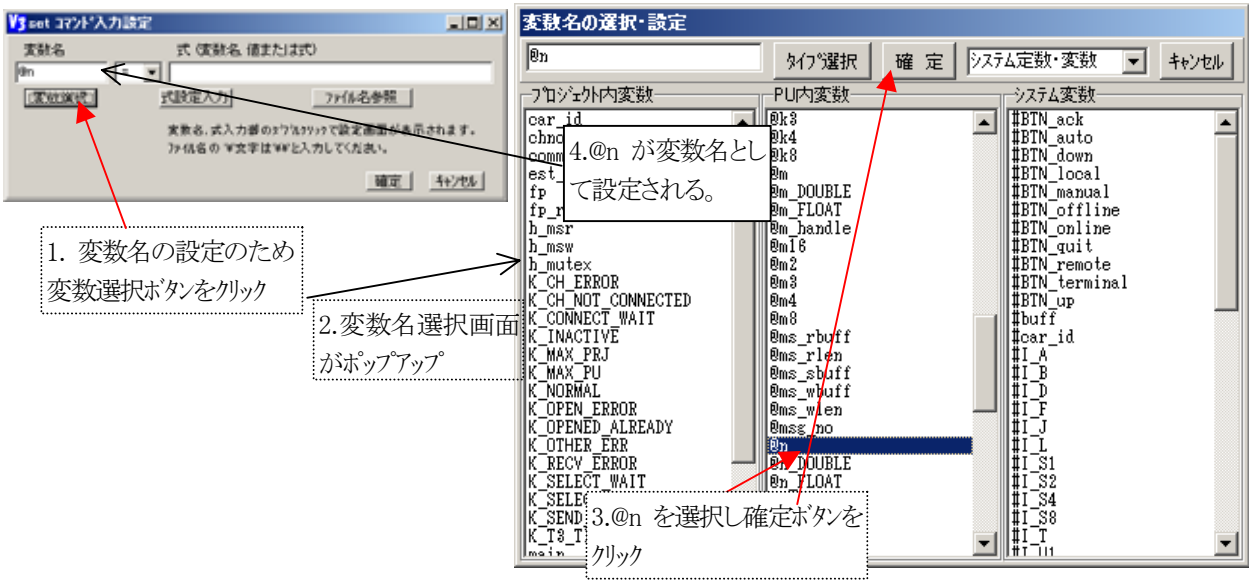
（4）-3 一般的にプログラミングしたい場合は、画面を見ながら以下のように作成することができます。

1. 1 行目のコメントはそのままセルにキー入力します。
2. 2 行目のラベルはそのまま start_s5f1 と入力します。
3. 3 行目の set コマンドは command 欄にそのまま set と入力してもいいですが、**ｺﾏﾝﾄﾞ選択**コンボボックスで set コマンドを選択し、**追加**ボタンをクリックすることによって設定できます。コマンド名やコマンドの形式を忘れた場合、このようにすれば、コマンド名とそのパラメータの内容を知ることができます。

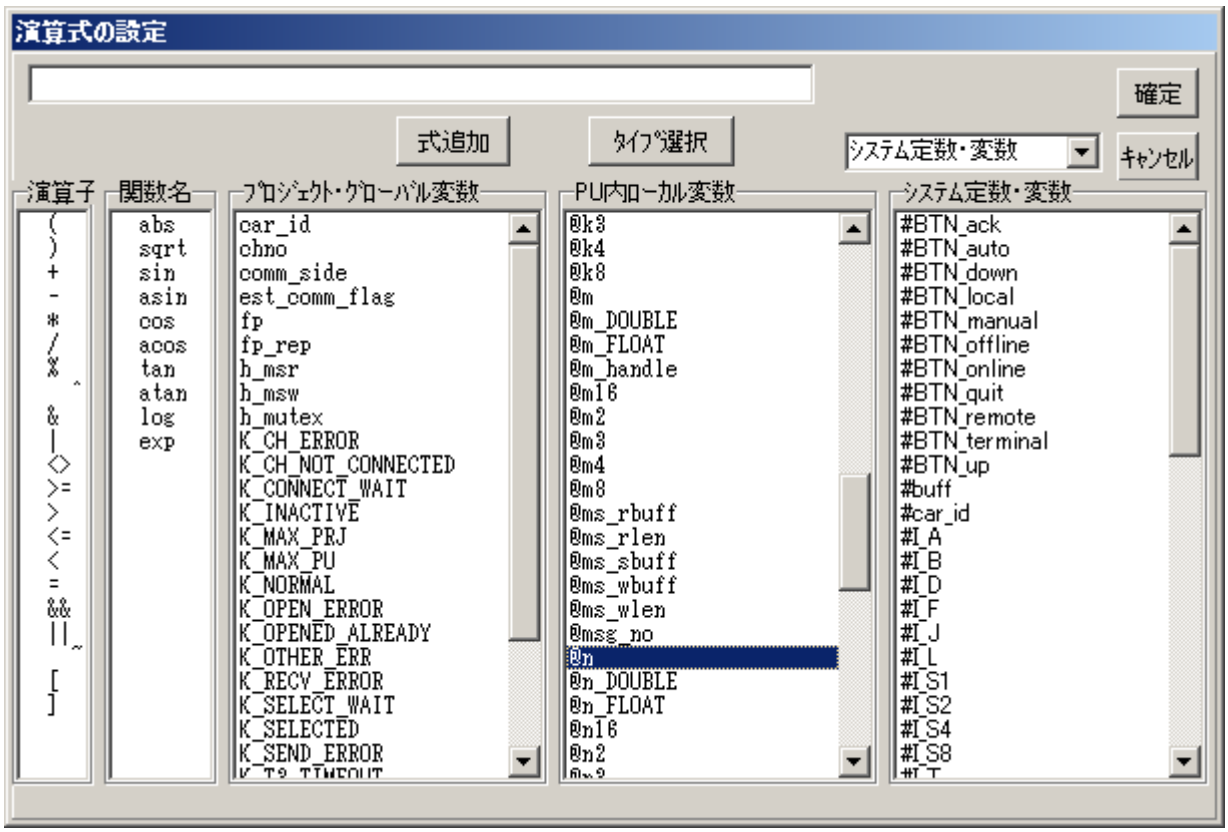


次に para-1 の[変数=式]の設定は、直接入力でもできますが、演算式を画面で入力設定することもできます。

他の方法として、set コマンドの para-1 のセルをダブルクリックすると以下の画面が出てきますので、ここで変数名、式を画面操作で生成し設定することもできます。



式の設定入力ボタンのクリックで式の入力設定画面が表示されます。ここでも変数と演算記号を使って式を作り確定ボタンをクリックします。



結果として、para-1 の欄を @n=0 にします。

4. 4 行目の inputkey コマンド以下のコマンドを入力設定して行きます。
   例外はありますが、パラメータを設定する必要があるところ、para-1 または para-2 のセルを
   ダブルクリックすることによって関連する設定画面がポップアップされ、一覧表画面をみながら
   選択設定することができます。

（5）シナリオプログラムの保存、保存形式と再ロード

画面上に作成したプログラムの保存は、メニューの **保存(S)** または **新規保存(A)**のクリックで行います。
シナリオファイルの拡張子は .seq です。

ファイル内の保存形式は、例えば、（4）の S5F1 送信テストプログラム例の場合、以下のようになります。

1. 各セル間はカンマ(,)で区切って各フィールドの内容が保存します。
2. コマンドのパラメータ内のカンマ(,)文字は ¥s で置き換えて保存します。
   （ロード時に ¥s をカンマに変換し戻します。）

```
//S5F1,,,,S5F１を２回送信して停止
start_s5f1,set,@n=0,,送信回数 @n=0
loop1,,,,繰り返し開始位置
,inputkey,ALID を入力してください。,@buff1,ALID を入力
,strtoint,@i2,@buff1,文字列を数値変換 ==> @i2
,inputkey,ALTX を入力してください。,@buff2,ALTX を@buff2 に入力
,,,,
,put_msg,s5f1_alarm,,s5f1_alarm ﾒｯｾｰｼﾞを put_item の対象にする。
,put_item,alid,@i2,ﾃﾞｰﾀｱｲﾃﾑ alid の値を @i2 にする。
,put_item,altx,@buff2,ﾃﾞｰﾀｱｲﾃﾑ altx の値を @buff2 にする。
,,,,
,send,s5f1_alarm,@seq_rmsg2,s5f1_alarm を送信する。
,get_msg,@seq_rmsg2,,s5f2 を get_item の対象ﾌﾟﾛﾄﾀｲﾌﾟにする。
,get_item,ackc5,@i3,@seq_rmsg2 から ackc5 を@i3 に取得する。
,outlog,"ackc5=%d"¥s@i3,,ackc5
,,,,
,set,@n=@n+1,,送信ｶｳﾝﾄ+1
,ifgo,@n<2,loop1,< 2 ならば繰り返し(loop1 へ)
,stop,,,>=2 ならば 停止(終了)
```

保存されたファイルは、メニューの **開く(O)** または **開きなおす(R)**のクリックでファイルを選択し画面に再ロードさせることができます。

（6）プログラムの編集

プログラムの編集は編集ボタンを使って行います。
コマンドの設定は、**追加**、**挿入**、**置換**ボタンを使って行います。
１行追加したい場合は、**空行追加**ボタンを使って１行分のブランク行を作ってコマンドを追加すると便利です。
削除ボタンは１行分のコマンドラインを削除し、そのライン以降のコマンドを前の方に詰めます。
ｺﾏﾝﾄﾞｸﾘｱボタンはその行の全カラムの表示をブランクにします。
コマンドパラメータの設定には、**パラメータのセルをダブルクリックする**ことによって各種情報の設定画面をポップアップさせて設定することができます。

- 変数名、演算式
- 定義メッセージ、メッセージ変数、メッセージのデータアイテム名
- ラベル名（一覧表）
- ファイル名とファイル名が格納されている変数（一覧表）
- シナリオファイル名（loadgo ｺﾏﾝﾄﾞ）

（7）シナリオプログラムの実行

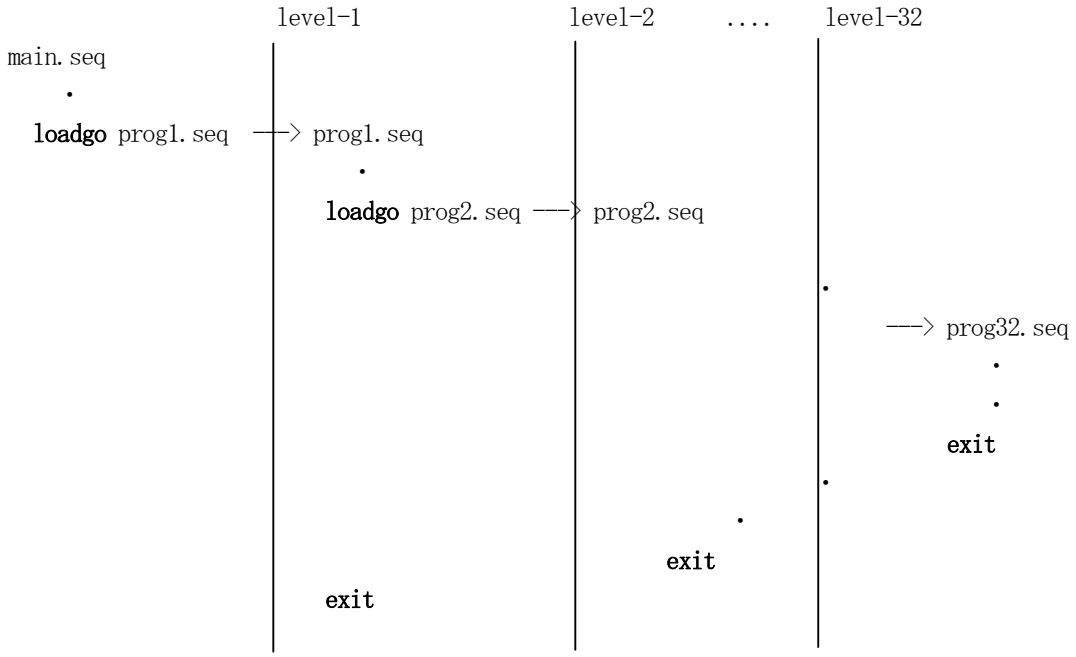
プログラムを最初から実行する際はまず、実行操作の**ﾘｾｯﾄ**ボタンをクリックして、シナリオ実行のための内部情報を初期化します。
このとき、プログラムの１行目のコマンドが選択されます。（最初に実行される行です。）
次に、**実行**ボタンをクリックします。実行中は画面上は、実行されているコマンド行が順次選択表示されます。

実行するステップの順番とコマンドを目で追跡しながらゆっくりテストをするために、コマンドとコマンドの実行の間に故意に遅延時間をとることもできます。**実行間隔(ms)**のコンボボックスの遅延時間選択によってそれが実現させることができます。

シミュレータは、コマンド実行時にエラーを検出した場合、エラーを起こしたステップ、コマンド名を表示し、実行を自動停止させます。

（8）サブルーチンコールとシナリオプログラムロードのネスティングの最大レベル

1. call コマンドによるサブルーチンのネスティングは最大３２レベルまで可能です。
2. loadgo コマンドによるシナリオプログラムのロードと実行のネスティングは最大３２レベルまでです。



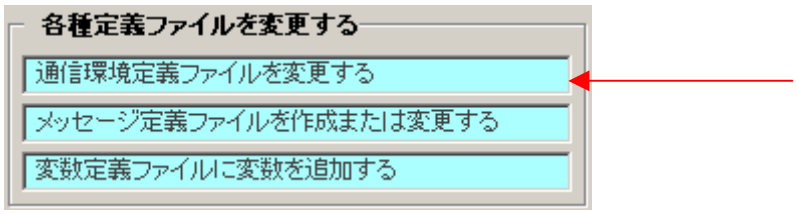
５．　通信環境定義ファイルの再定義

通信環境定義ファイル（COMM.DEF）の内容の設定変更のための編集とファイル保存を行うための操作です。
基本的には、DSHSIM.EXE シミュレータの起動パラメータで指定された通信環境定義ファイル内に指定された情報に対する変更と保存になります。

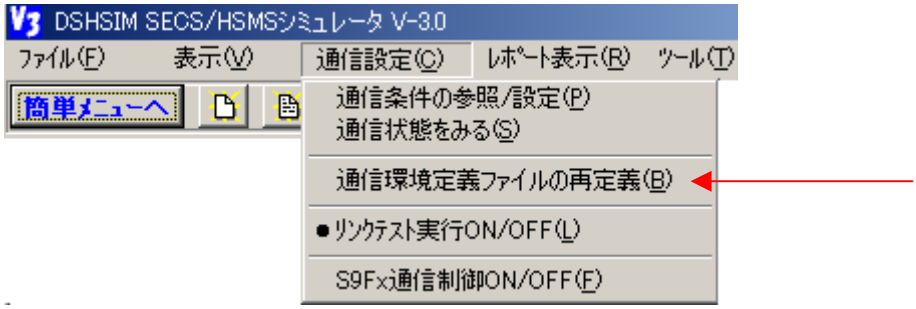
（1）操作の開始

操作は、次のどちらかの操作で始めます。

① 簡単メニュー

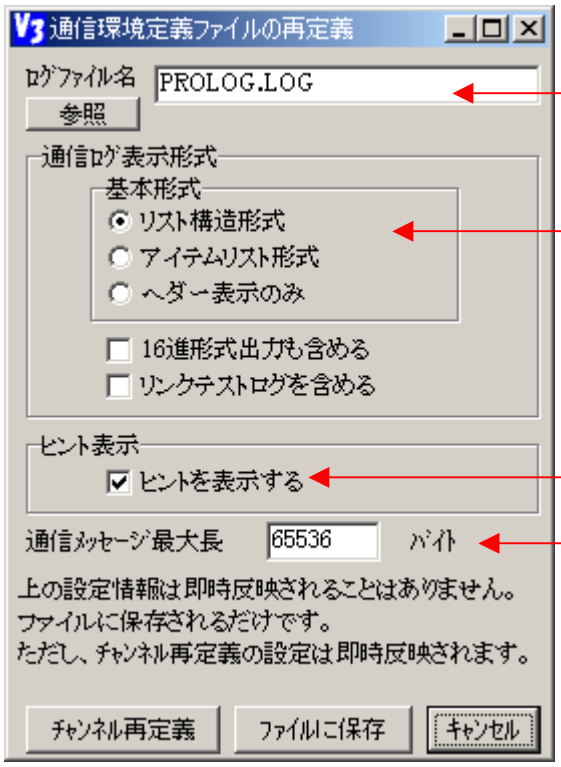


② メイン画面の通信設定(C)ﾒﾆｭｰの通信環境定義ﾌｧｲﾙの再定義(B)ﾀﾌﾞ



（2）操作画面

操作を開始すると、設定変更するチャンネル選択のための画面が表示されます。
各項目には、現設定値が表示されます。



ﾛｸﾞﾌｧｲﾙ名を変更することができます。

通信ﾛｸﾞ表示形式を変えることができます。

ﾎﾞﾀﾝなどのﾋﾝﾄ表示を “する”／”しない”を選択します。

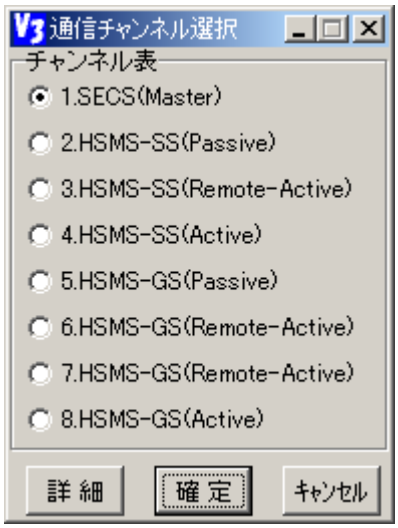
ﾌﾟﾛﾄｺﾙ上通信できるﾒｯｾｰｼﾞの最大長を指定します。
受信時、この値を超えるﾒｯｾｰｼﾞ受信した場合、HSMS-SSﾌﾟﾛﾄｺﾙﾁｬﾝﾈﾙは通信が一旦切断され、再接続されることになります。

各項目の設定が済んだら、**ﾌｧｲﾙに保存**ボタンをクリックし、ファイル保存画面でファイル名を指定して保存します。
ﾁｬﾝﾈﾙ再定義ボタンについては、次の（3）で説明します。

（3）チャンネル再定義

ﾁｬﾝﾈﾙ定義ボタンの機能で、チャンネルのプロトコル変更を含めたチャンネルの再定義を行うことができます。

① このボタンをクリックすると、チャンネル選択画面が表示されます。



ここで、再定義したいﾁｬﾝﾈﾙを選択し、
確定をクリックします。

（注）
もし、指定ﾁｬﾝﾈﾙがﾌﾟﾛｼﾞｪｸﾄ実行によって使用中の場合は、操作が拒否されます。

② 選択チャンネルのプロトコル選択画面が表示されます。

ここで､ﾌﾟﾛﾄｺﾙを選択し､確定をｸﾘｯｸします。

③ この後は、指定されたプロトコルの通信チャンネル情報の設定変更操作を行います。
詳細操作はこちらを参照してください。 通信制御パラメータの設定変更

ここで、再定義されたチャンネル情報は、ただちに、シミュレータ内部に反映され、次にそのチャンネルが使用されたときは、再定義された内容で、通信が行われます。

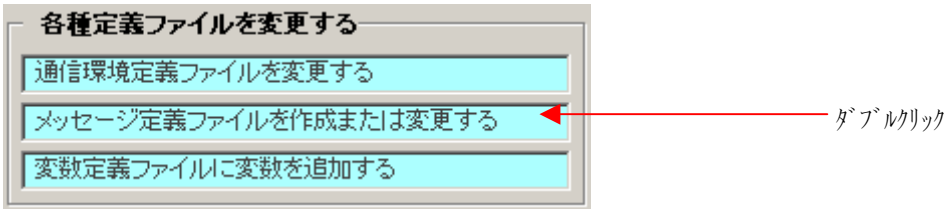
６．　メッセージファイルの作成と変更

通信メッセージ定義ファイル（.MSG）作成と変更のための編集ならびに保存するための操作です。

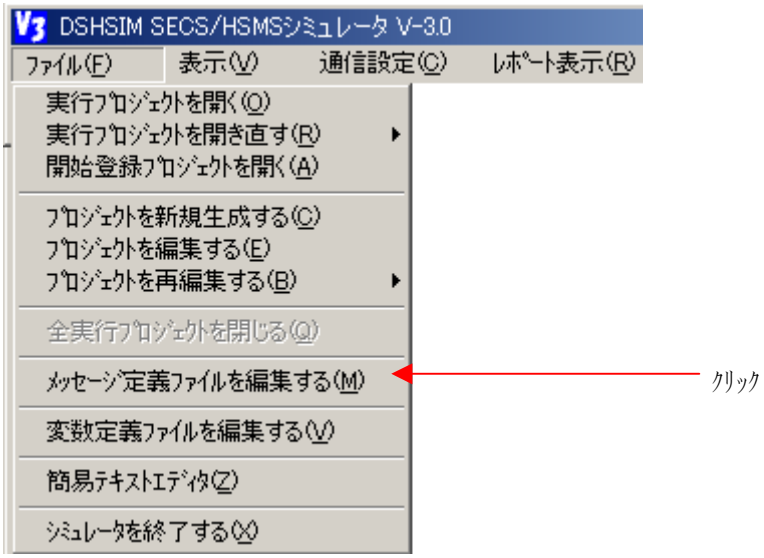
（1）操作の開始

操作は、次のどちらかの操作で始めます。

① 簡単メニュー

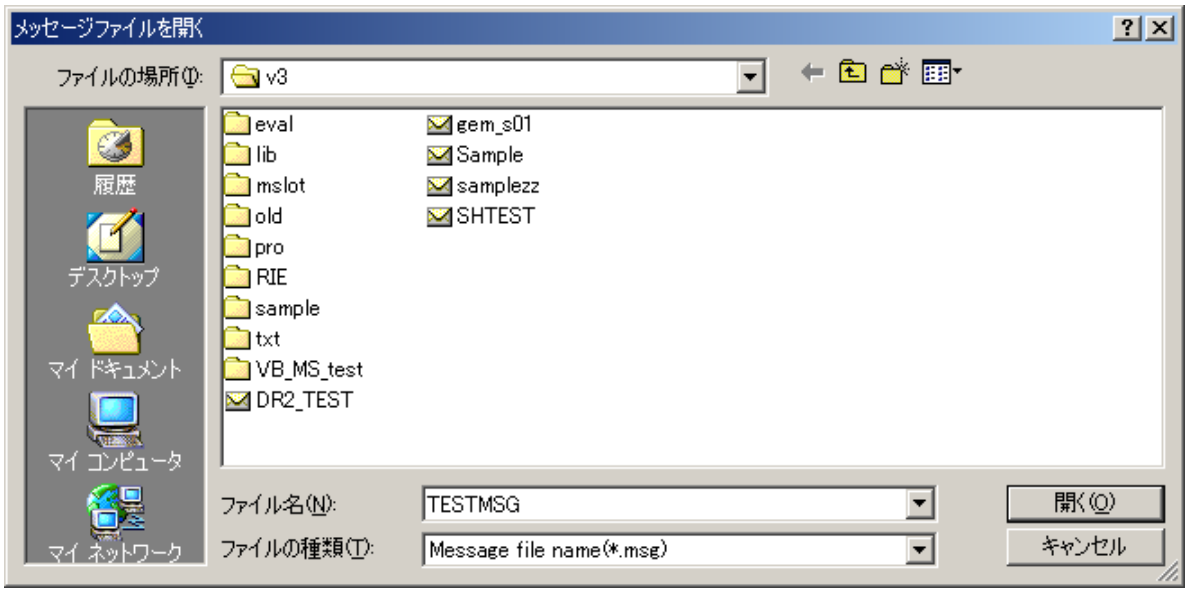


② メイン画面のﾌｧｲﾙ(F)ﾒﾆｭｰのﾒｯｾｰｼﾞ定義ﾌｧｲﾙを編集する(M)ﾀﾌﾞ



（2）ファイル名指定画面

操作を開始すると、まず、最初に編集するメッセージ定義ファイル名の選択画面が表示されます。



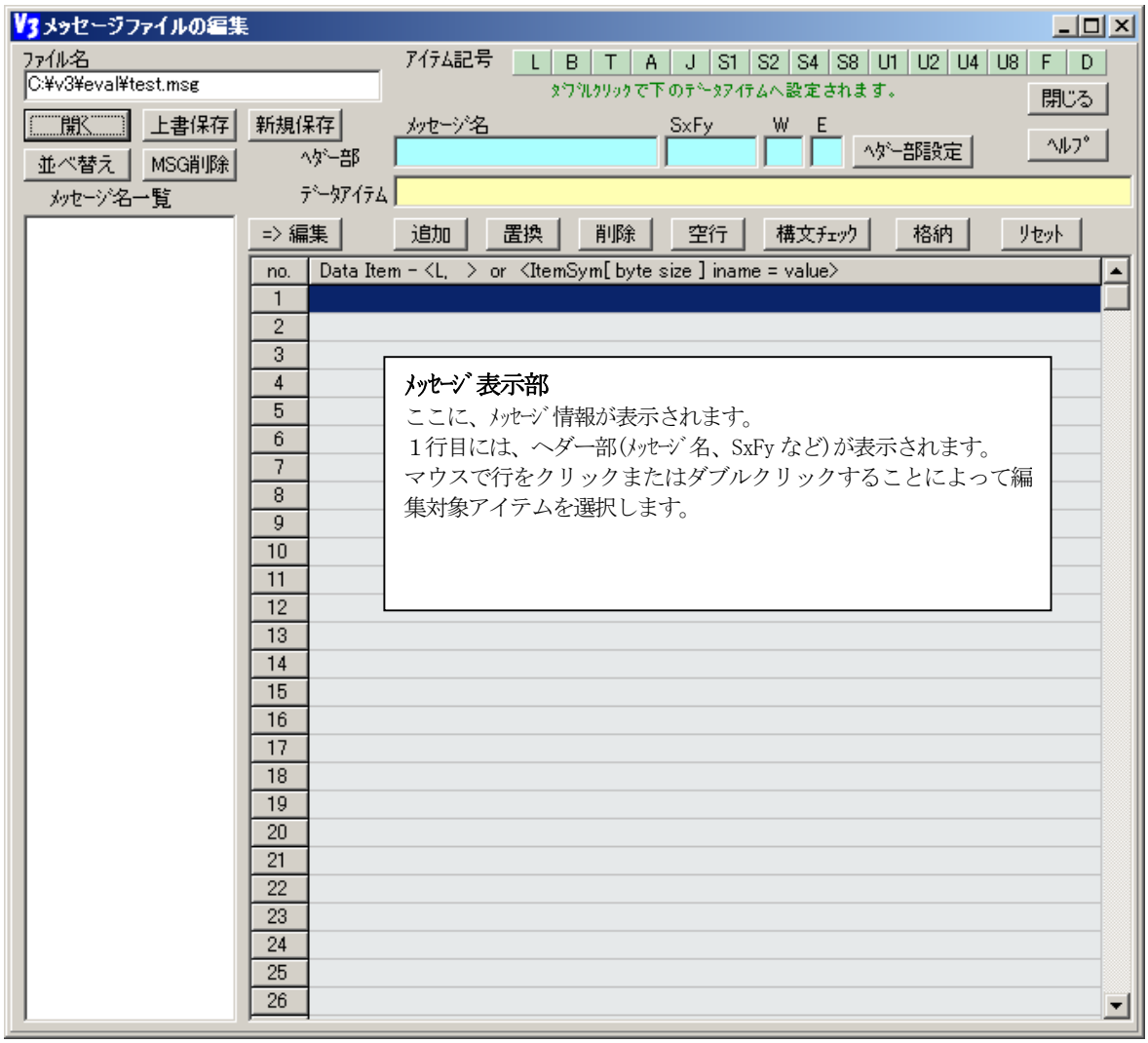
① 新しくメッセージファイルを作成したい場合は、ここで、その名前を入力します。
② もし、既存のメッセージファイルを編集する場合は、ファイル名を選択します。
ファイル名を指定したあと、**開く**ボタンをクリックします。

以下、例として、TESTMSG.MSG ファイルを新規作成する操作について説明します。

（3）メッセージ編集操作画面

最初、次の画面がポップアップ表示され、新規作成であることを知らせます。

次に、以下の編集画面が表示されます。



大まかな操作はつぎのようになります。

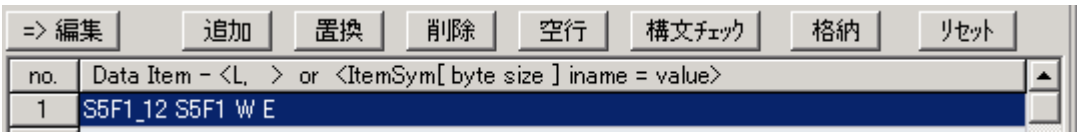
（4）S5F1 メッセージを作成するための操作

次の S5F1_12 メッセージを作成し、シミュレータに追加します。

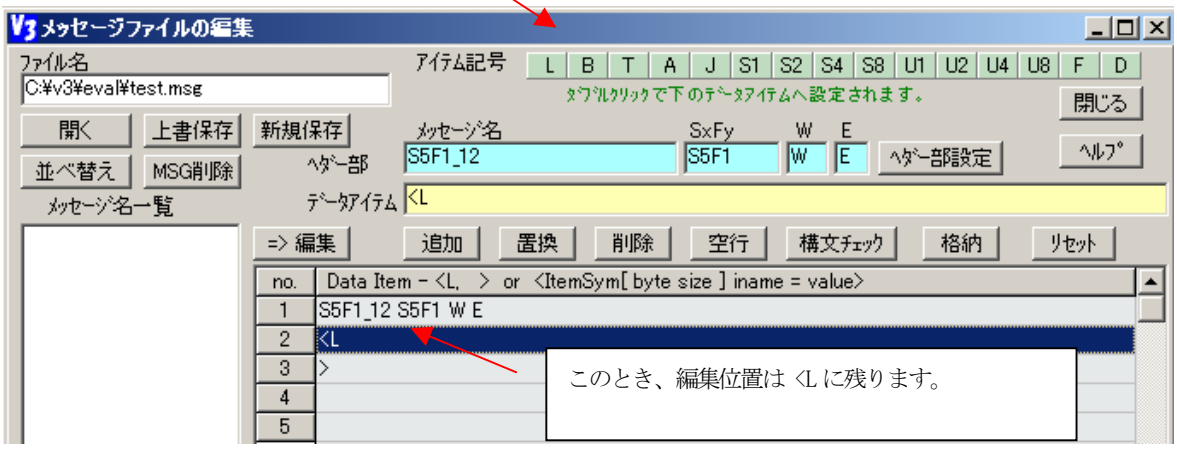
```
S5F1_12 S5F1 W E
<L
  <B[1] alcd = x8c>
  <U2[2] alid = 1>
  <A[40] altx = “Carrier Sensor Error”>
>
```

① まず、メッセージ名、メッセージID、W-BIT、通信サイドの情報をヘダー部に設定します。
それぞれ、以下のようにキー入力し、**ﾍﾀﾞｰ部設定**ボタンをクリックします。

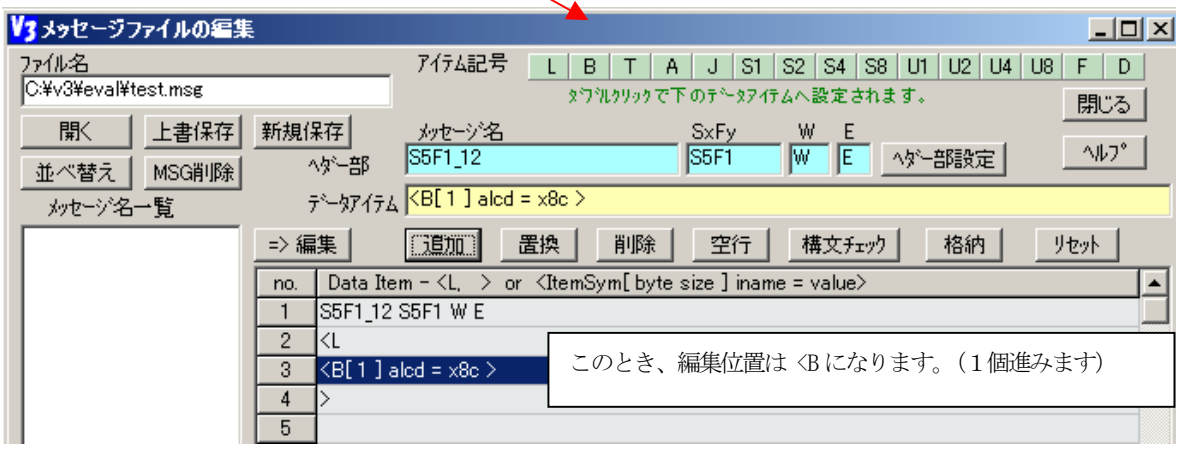
ﾒｯｾｰｼﾞ表示部の１行目に ”S5F1_12 S5F1 W E” と下図のように表示されます。



② 以下、アイテムデータの設定になります。最初のアイテム <L を追加します。
・ｱｲﾃﾑ記号 L をﾀﾞﾌﾞﾙｸﾘｯｸします。
ｱｲﾃﾑﾃﾞｰﾀ部に、 <L が表示されます。
・追加ボタンをクリックします。
ﾒｯｾｰｼﾞ表示部に<L, > が表示されます。これで、List アイテムの設定終わりです。



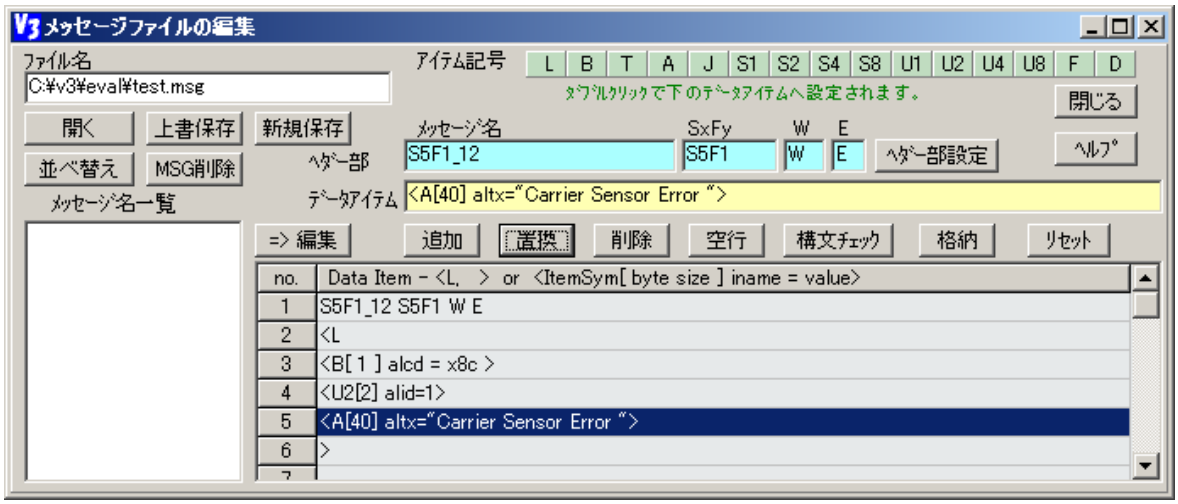
③ <B[1] alcd = 12>を設定します。
・ｱｲﾃﾑ記号 B をﾀﾞﾌﾞﾙｸﾘｯｸします。
ｱｲﾃﾑﾃﾞｰﾀ部に、 “<B[1] = 0 >” と表示されます。
この内容を alcd=x8c とキー入力で変更します。
・追加ボタンをクリックします。これで、<B[1] . . . >がﾒｯｾｰｼﾞ表示部に追加されます。



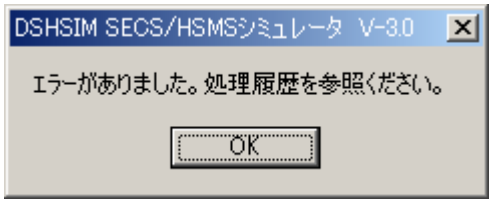
④ <U2[2] alid = 1>を設定します。

・ｱｲﾃﾑ記号 U2 をﾀﾞﾌﾞﾙクリックします。
ｱｲﾃﾑﾃﾞｰﾀ部に、“<U2[2] = 0 >” と表示されます。
この内容を alid=1 と変更します。
・追加ボタンをクリックします。これで、<U2[2] ... >がﾒｯｾｰｼﾞ表示部に追加されます。

⑤ 最後に、<A[40] altx = “Carrier Sensor Error”>を設定します。
・ｱｲﾃﾑ記号 A をﾀﾞﾌﾞﾙクリックします。
ｱｲﾃﾑﾃﾞｰﾀ部に、“<A[1] = 0 >” と表示されます。
この内容を altx = “Carrier Sensor Error”と変更します。
・追加ボタンをクリックします。これで、<A[40] ... >がﾒｯｾｰｼﾞ表示部に追加されます。
結果、画面は次のようになります。

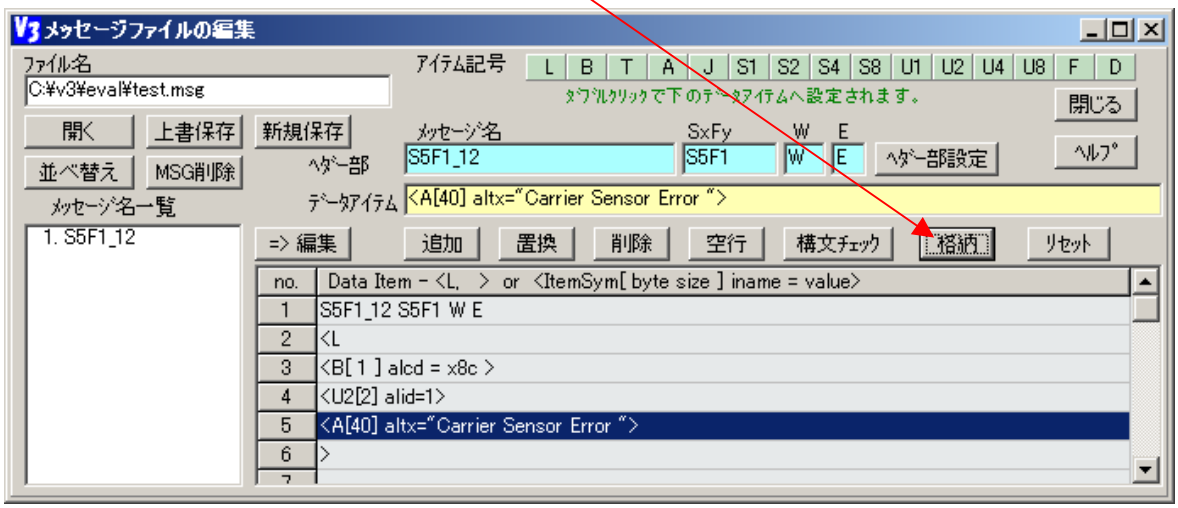


⑥ 次に、設定したメッセージの構文が正しいかどうかを**構文ﾁｪｯｸ**ボタンのクリックで行います。
チェック結果によって次のように表示されます。



ｴﾗｰが検出された場合、
処理履歴画面にｴﾗｰ個所と
思われる場所が表示されます。

⑦ チェックが終わったら,**格納**ボタンをクリックしてシミュレータ内部に記憶します。
結果、メッセージ名一覧に S5F1_12 の表示が追加されます。
また、“格納しました。”のポップアップ画面が表示されます。

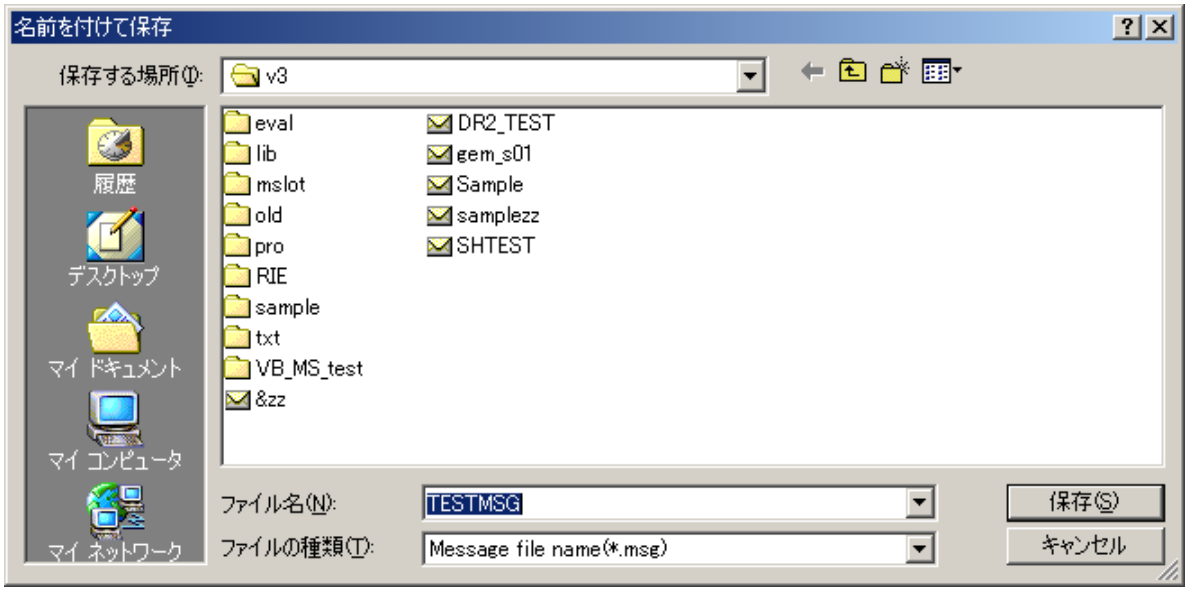


以上で、S5F1_12　１個を追加することができました。(ただし、ここでは、まだファイルには保存されていません。)

以下、必要なメッセージを順に作成し、格納することになります。
次のメッセージを作成する前に、ﾘｾｯﾄボタンで、前のメッセージの設定内容をクリアすることができます。

（5）格納したメッセージのファイルへの保存

作成したメッセージの保存は、画面左上の**保存**ボタンで行います。

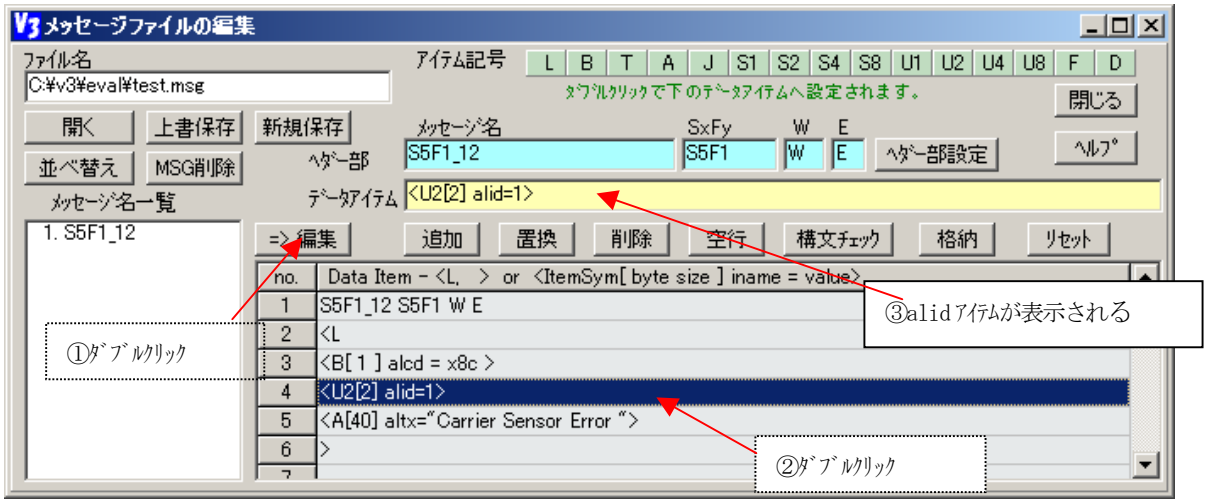


名前を TESTMSG.MSG で保存する場合は、そのまま保存してください。
（注）本操作でメッセージ情報をファイルに保存すると';'文字で始まるコメントは全て消失　します。

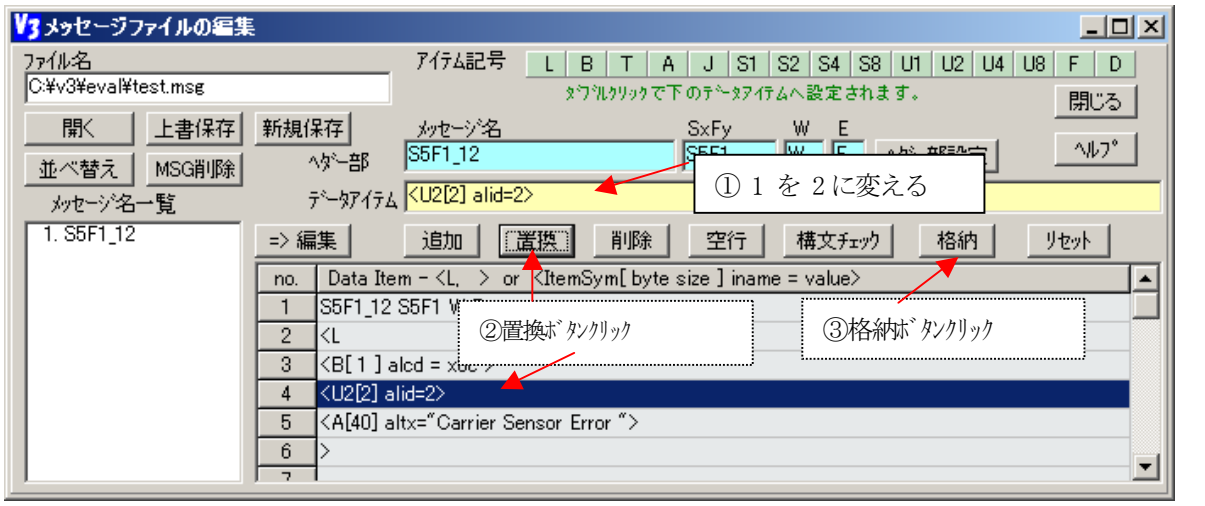
これで、一通りの作成操作の終わりです。
この後、S5F2 メッセージも作成したことにして、既存のメッセージの変更操作について説明します。」

（6）メッセージ内アイテム変更操作

例えば、先に作成した S5F1_12 の alid アイテムの値を５に変更するための操作は次のように行います。
① メッセージ名一覧の中の S5F1_12 をダブルクリックします。右側の画面に S5F1_12 の情報が表示されます。
② 次に、ﾒｯｾｰｼﾞ表示部の４行目 <U2[2] alid=1> をダブルクリックします
③ その内容が上のｱｲﾃﾑﾃﾞｰﾀ部に表示させます。



④ 次に、そのｱｲﾃﾑﾃﾞｰﾀの値 alid = 1 を alid = 2 にキー入力で変更します。
⑤ その後、**置換**ボタンをクリックし、ｱｲﾃﾑﾃﾞｰﾀ部の内容をﾒｯｾｰｼﾞ表示部の４行目に上書きします。
⑥ そして、**格納**ボタンのクリックによって、シミュレータ内部に格納します。



（7）アイテムデータの追加

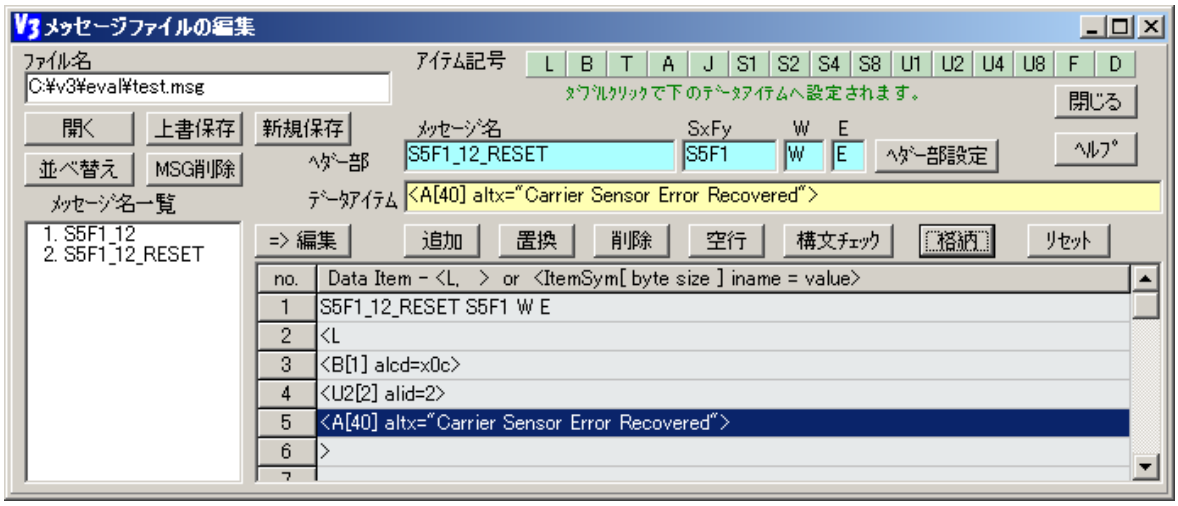
アイテムの追加操作は、（4）で述べた S5F1_12 のアイテムデータの操作と同様に行います。ただし、最初にどの位置に追加するかを選択します。
たとえば、5 行目の〈A40〕altx=”....”>の後に、追加する場合は次のように操作します。
① 最初に、ﾒｯｾｰｼﾞ表示部の 5 行目をクリックし、編集位置を指定します。
② 次に、アイテム記号のクリックでアイテムデータ部にアイテムデータの基本形をセットします。
③ サイズ、アイテム名、値をセットし、**追加**ボタンをクリックします。
これで、新しい、アイテムが 6 行目に追加されました。元の 6 行目以降のデータは下方にシフトダウンされます。

（8）アイテムデータの削除

アイテムデータの削除は簡単です。
① 最初に、ﾒｯｾｰｼﾞ表示部の削除したいアイテムデータの行をマウスで選択します。
② その後、**削除**ボタンをクリックします。削除行以降のアイテムデータが上にシフトアップされます。

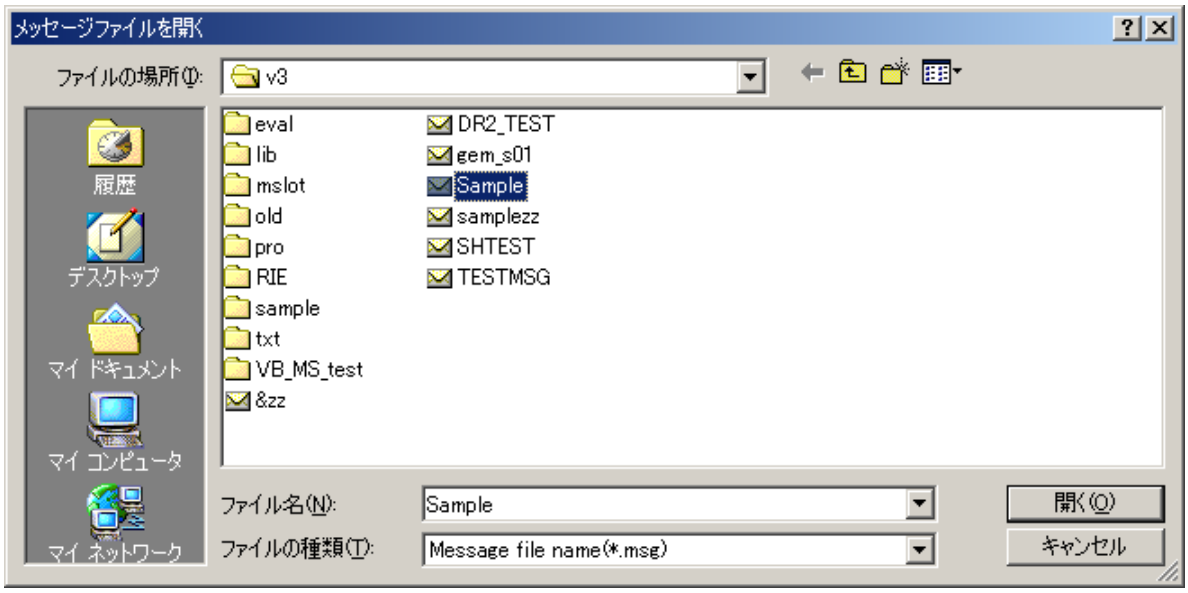
（9）既存メッセージから別名メッセージへの作成

例えば、既存メッセージ S5F1_12 を基に、alcd と altx を変えて S5F1_12_RESET メッセージを作りたい場合があります。
操作は次のように行います
① ﾒｯｾｰｼﾞ一覧の **S5F1_12** をダブルクリックし、メッセージ情報をﾒｯｾｰｼﾞ表示部に表示します。
② まず、ﾒｯｾｰｼﾞ名を **S5F1_12_RESET** に変更し、**ﾍﾀﾞｰ部設定**ボタンをクリックします。
③ 表示部の〈B[1] alcd=x8c〉をダブルクリックし､alcd のアイテムをアイテムデータ部に表示させます。
ここで x8c を x0c に変更し、**置換**ボタンをクリックします。これで、alcd の変更が済みました。
④ 次に、③と同様に、〈A[40] alcd=”．．　“〉内容を変えます。
⑤ 最後に、**格納**ボタンをクリックし、内部に記憶させます。
操作の結果、画面は次のようになります。S5F1_12__RESET メッセージが追加されました。この後、ファイルに保存してください。



(10) 別のメッセージファイルの編集操作

１つのメッセージファイルの編集が終わり、ファイルに保存したあと、別のメッセージファイルを編集する場合は、左上隅の**開く**ボタンをクリックします。
次の“メッセージファイルを開く”画面が表示されますので、ここで、メッセージファイル名を選択し、**開く**ボタンをクリックしてください。
そのあとは、（6）～（9）の操作でメッセージを変更することができます。


メッセージファイルを開く
ファイルの場所(I):
v3
履歴
デスクトップ
マイ ドキュメント
マイ コンピュータ
マイ ネットワーク
eval
lib
mslot
old
pro
RIE
sample
txt
VB_MS_test
&zz
DR2_TEST
gem_s01
Sample
samplezz
SHTEST
TESTMSG
ファイル名(N):
Sample
ファイルの種類(T):
Message file name(*.msg)
開く(O)
キャンセル

7． 変数定義ファイルの編集

変数定義ファイル(variable.txt)への新しい変数の追加または、既存変数定義の変更、削除などの編集操作を行います。

本操作によって、ユーザがプログラム内で使用したい新しい変数を定義ファイルに追加保存することができます。

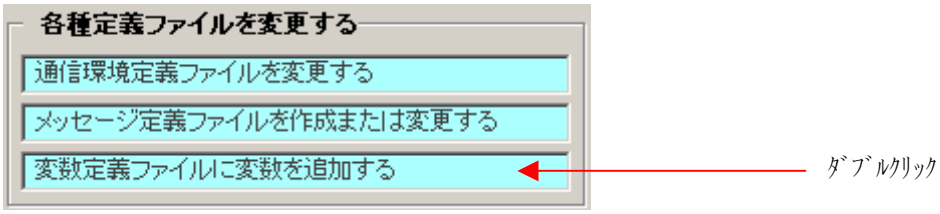
（注） 自分で新規に定義した変数以外の変数を削除しないようにしてください。

編集操作の結果は、即時、シミュレータ内部に反映されます。

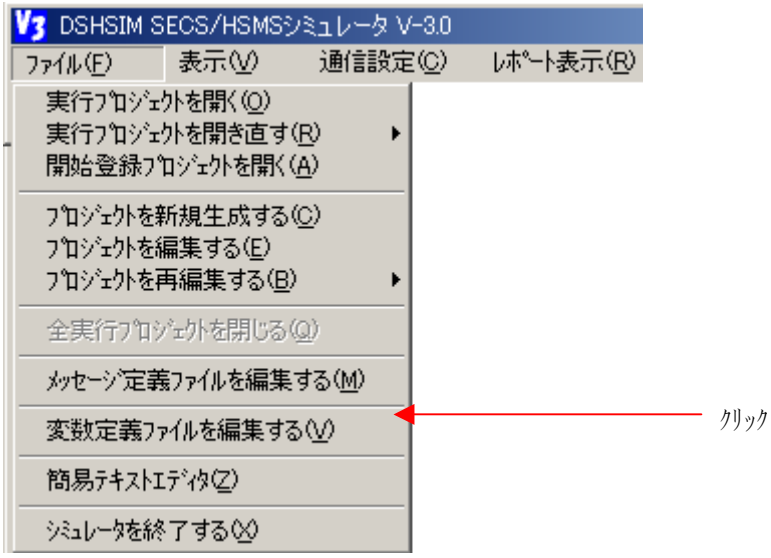
（1）操作の開始

操作は、次のどちらかの操作で始めます。

① 簡単メニュー



② メイン画面のﾌｧｲﾙ(F)ﾒﾆｭｰの**変数定義ﾌｧｲﾙを編集する(V)**ﾀﾌﾞ



（2）編集画面

次の編集画面が表示されます。

画面の下側には、現在定義されている変数一覧表が表示されます。

① **ﾌﾟﾛｼﾞｪｸﾄ内変数**は、プロジェクト内のグローバル変数です。

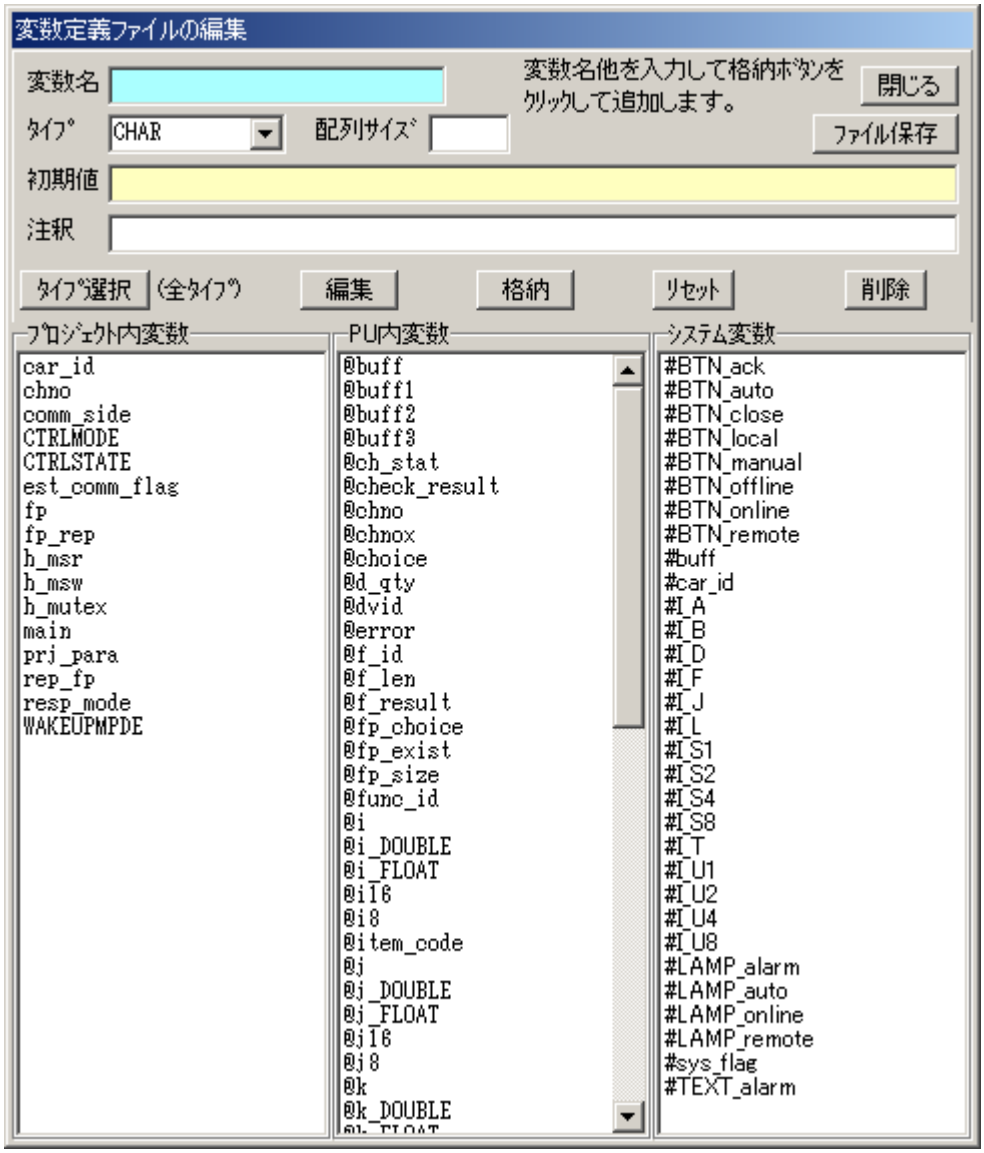
プロジェクト内に、変数の実体が１個だけ存在します。プロジェクトのどのPUからもアクセスできます。

② **PU内変数**は、PU内のローカル変数です。各PU内に独立した実体があります。

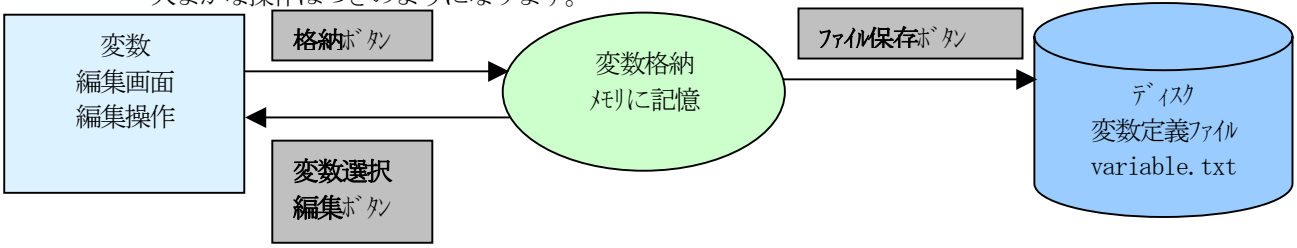
自PUだけがアクセスでき、他のPUからはアクセスすることができません。名前の先頭文字は @ 文字で始まります。

③ **ｼｽﾃﾑ変数**は、シミュレータ内に１個だけ実体があります。

どのプロジェクト、どのPUからもアクセスできます。名前の先頭文字は、 # 文字で始まります。



大まかな操作はつぎのようになります。

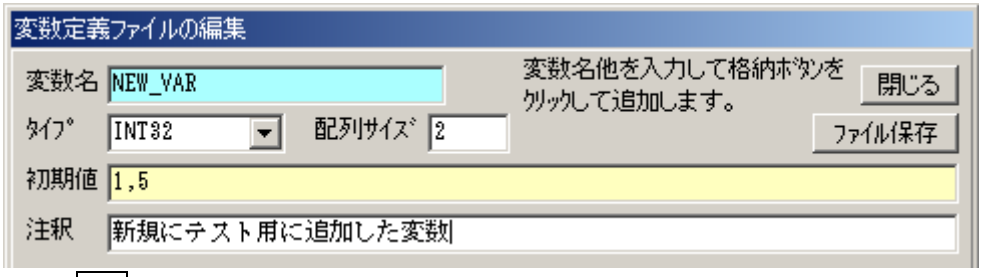


（3）新規に変数を定義する

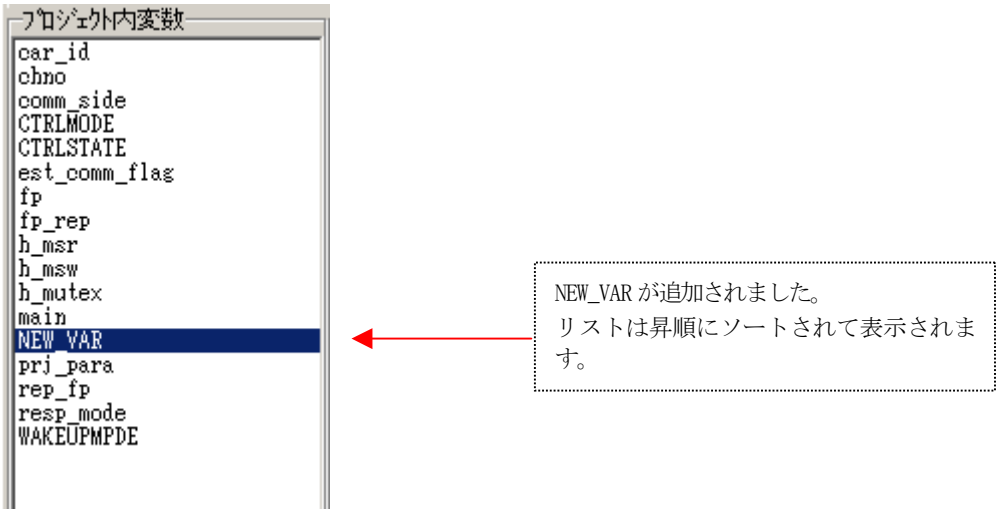
新規に変数名 次のプロジェクト内変数を定義するための操作について順を追って説明します。

変数名　　：　NEW_VAR
タイプ　　：　INT32
配列サイズ　：　2
初期値　　：　NEW_VAR[0] = 1,　NEW_VAR[1] = 5
注釈　　：　“新規にテスト用に追加した変数”

① 変数名、タイプの選択、配列サイズ、初期値、注釈の項目を下図のように設定します。
初期値は、配列の場合は、カンマ切りで値を設定してください。(サイズ分すべてを設定しなくても構いません。)



② 次に、格納ボタンをクリックします。
結果、画面は次のようになります。



③ その後、ﾌｧｲﾙ保存ボタンクリックで、新しい変数を含めて全変数が、variable.txt ファイルに保存されます。

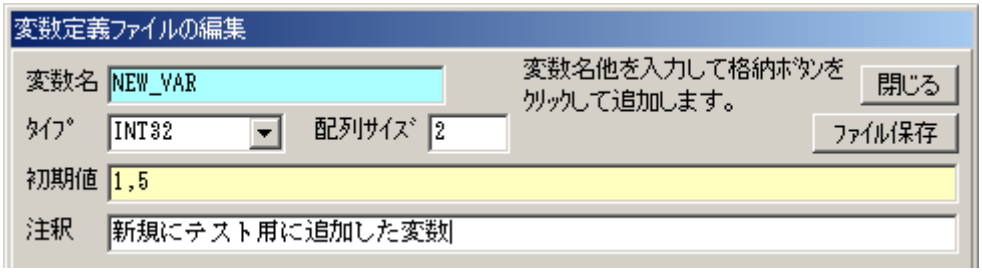
（4）既存変数の定義変更

既存変数の定義内容変更は、変数リスト上にある目的の変数名を選択し、編集ボタンをクリックすることによってできます。

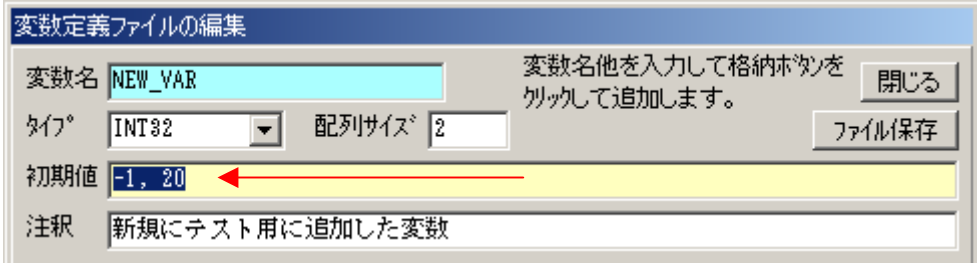
（目的の変数名のダブルクリックでも編集を開始できます。）

例えば、（3）で新規定義した NEW_VAR の初期値 1,5 を -1, 20 に変更するための操作は次のようになります。

① ﾌﾟﾛｼﾞｪｸﾄ内変数一覧表の中の NEW_VAR をダブルクリックします。画面は次のようになります。
（先に、定義した内容がそのまま表示されます。）



② 初期値 1, 5 を キー入力で -1, 20 に変更します。



③　この後、格納ボタンクリックで変更内容が内部に取り込まれます。

（5）既存変数の削除

削除操作は簡単です。

①　まず、削除したい変数名をクリックし、選択します。

②　次に削除ボタンをクリックします。

例えば、先ほど定義した NEW_VAR の削除を例にすると、確認画面が表示されます。

ここで、”はい”のクリックで削除完了です。

削除された変数名は表示から消えます。

（6）ファイルへの保存

ﾌｧｲﾙ保存ボタンクリックで、新しい変数を含めて全変数が、variable.txt ファイルに保存されます。

（注）本機能を使ってファイル保存すると、// “文字”で始まるコメントは消失します。

８．　通信環境定義ファイル仕様

シミュレータは、起動時に動作するための環境情報を通信環境定義ファイル COMM.DEF から読出し、参照しながら動作を行います。(起動パラメータとして、通信環境ファイル名が指定されれば、それを参照します。)

環境定義ファイルのデフォルト名は COMM.DEF と決められています。DSHSIM.EXE シミュレータプログラムの起動パラメータとして任意のファイル名を使用することができます。

通信環境定義ファイルは、テキストファイルであり、その中には、通信チャンネルの設定をはじめとする幾つかの種類のコマンドを使って環境情報を定義します。

［ バージョン－２．０とバージョン－３．０の違い］
バージョン－３．０では、以下のコマンドは使用いたしません。

コマンドの種類	コマンド名	使用しない理由
1.　ﾏﾆｭｱﾙﾁｬﾝﾈﾙ関連	MANSIDE MANCH	ﾊﾞｰｼﾞｮﾝ-3.0 では、ﾏﾆｭｱﾙ、ｵｰﾄ通信の区別がなく、また、また各ﾁｬﾝﾈﾙについて独立的な通信操作のためのﾊﾟﾈﾙがある。そのため、必要なくなった。 （ﾊﾞｰｼﾞｮﾝ-2.0 では１つの操作ﾊﾟﾈﾙで通信操作を行っていた。）
2. ﾒｰﾙｽﾛｯﾄ関連	MSNAMEi	ﾊﾞｰｼﾞｮﾝ-3.0 では、ﾌﾟﾛｸﾞﾗﾑｺﾏﾝﾄﾞを使用して自由にﾒｰﾙｽﾛｯﾄのｵﾍﾟﾚｰｼｮﾝができる。

本章では、環境定義コマンドとその使い方について説明します。

８．１　コマンド形式とコマンド一覧表

基本的なコマンドの形式は次のとおりです。

<コマンド> = ［<パラメータ>］

種類別に以下の定義コマンドが準備されています。

コマンド一覧表

番号	コマンド名	パラメータ	機　能
1.	START	<CHNO>	<CHNO>の制御情報の設定開始
2.	END	なし	<CHNO>の制御情報設定終了
3.	SECS	<MSFLAG>	SECS であり、MASTER/SLAVE の識別設定
4.	HSMS	<Entity>	HSMS であり、PASSIVE/ACITIVE/REMOTE の識別設定
5.	SESSION	<Senssion>	HSMS の SS か、GS かを指定する。
6.	DVID	<DeviceID>	ﾃﾞﾊﾞｲｽ ID(ｾｯｼｮﾝ ID)の設定（16 進表現で設定）
7.	PORT	<COMi or Addr>	Serial COMM ﾎﾟｰﾄまたは TCP ﾎﾟｰﾄの設定
8.	BAUD	<BAUDRATE>	SECS の場合のﾎﾞｰﾚｰﾄの設定
9.	IP	<Address>	HSMS で接続相手 Active の TCP/IP のｱﾄﾞﾚｽを設定
10.	SVRNAME	<Address>	HSMS で接続相手 Passive の IP ｱﾄﾞﾚｽ設定(=SVRIP)
11.	DVRIP	<Address>	SVRNAME と内容は同じ（ﾊﾞｰｼﾞｮﾝでの互換性のため存在）
12.	CLIENT	<ANY/DEFINE>	HSMS Passive ﾎﾟｰﾄが接続する ACTIVE の制限
13.	DISPCH	<ACTIVE/PASSIVE>	HSMS Passive ﾎﾟｰﾄが表示する CH 番号の設定
14.	RETRY	<COUNT>	SECS のﾌﾟﾛﾄｺﾙﾘﾄﾗｲ回数の設定
15.	T1	<TVALUE>	SECS ﾌﾟﾛﾄｺﾙﾀｲﾏｰ T1 の設定（単位 100ms）
16.	T2	<TVALUE>	“　　T2　　“
17.	T3	<TVALUE>	SECS, HSMS ﾌﾟﾛﾄｺﾙﾀｲﾏｰ T3 の設定（単位 100ms）
18.	T4	<TVALUE>	SECS ﾌﾟﾛﾄｺﾙﾀｲﾏｰ T4 の設定（単位 100ms）
19.	T5	<TVALUE>	HSMS ﾌﾟﾛﾄｺﾙﾀｲﾏｰ T5 の設定（単位 100ms）
20.	T6	<TVALUE>	“　　T6　　“
21.	T7	<TVALUE>	“　　T7　　“
22.	T8	<TVALUE>	“　　T8　　“
23.	LINKTEST	<TINT>	HSMS で LINKTEST 実施する間隔（単位 sec）
24.	S9F0_RSP	<ON/OFF>	SECS でﾌﾞﾛｯｸ番号が間違っている場合の S9F0 送信設定
25.	S9F1_RSP	<ON/OFF>	ﾃﾞﾊﾞｲｽ ID が間違っている場合の S9F1 送信設定
26.	S9F9_RSP	<ON/OFF>	T3 ﾀｲﾑｱｳﾄ検出時の S9F9 送信指定
27.	S9F11_RSP	<ON/OFF>	MAX_S9F11_SIZE を超えるﾒｯｾｰｼﾞ受信時の S9F11 送信設定
28.	DVID_CK	<ON/OFF>	間違ったﾃﾞﾊﾞｲｽ ID のﾒｯｾｰｼﾞ受信時の無視設定
29.	WBLK_CK	<ON/OFF>	同一ﾒｯｾｰｼﾞの連続受信のﾁｪｯｸの設定
30.	MAX_S9F11_SIZE	<SIZE>	ﾁｬﾝﾈﾙが許容するﾒｯｾｰｼﾞ最大長(ﾊﾞｲﾄ)超えると S9F11 を送信する。(S9F11_RSP=ON の場合)ﾃﾞﾌｫﾙﾄ値は 65536 ﾊﾞｲﾄ
31.	LOGFILE	<FNAME>	処理履歴ﾌｧｲﾙ名の指定
32.	LOGFMT	<FMT>	通信ﾛｸﾞ表現形式の指定(FMT:LIST/ITEM/HEAD/HEX/CODE)
33.	LINKLOG	<ON/OFF>	HSMS ﾘﾝｸﾃｽﾄﾒｯｾｰｼﾞのﾛｸﾞ表示指定
35.	MSNAMEi	<NAME>	i 番目のﾒｰﾙｽﾛｯﾄ名の指定(i=1～8)（V3 では使用しない）
36.	HINT	<FLAG>	画面上のﾋﾝﾄ表示の選択指定
37.	MANSIDE	<HOST/EQUIPMENT>	ﾏﾆｭｱﾙ通信時のﾎｽﾄ/装置ｻｲﾄﾞのﾃﾞﾌｫﾙﾄ指定(V3 では使用しない)

38.	MANCH	<CHNO,CHNO,..>	ﾏﾆｭｱﾙ通信でｵｰﾌﾟﾝするﾃﾞﾌｫﾙﾄﾁｬﾝﾈﾙ(複数可能) (V3 では使用しない)
39.	MAX_LENGTH	<SIZE>	ｼﾐｭﾚｰﾀが許容するﾒｯｾｰｼﾞ最大長(ﾊﾞｲﾄ) これを超えるﾒｯｾｰｼﾞを受信時、HSMS の場合、接続を切断します。ﾃﾞﾌｫﾙﾄ値は 65536 ﾊﾞｲﾄ 最大値は 256K ﾊﾞｲﾄまで。
40.	CH_LOG	<ON/OFF>	通信ﾁｬﾝﾈﾙ別の通信ﾛｸﾞﾌｧｲﾙへの記録の指定

(注)

① 番号 1.～30. まではﾁｬﾝﾈﾙ定義用ｺﾏﾝﾄﾞです。

② 4. の HSMS=ACTIVE または HSMS=REMOTE ｺﾏﾝﾄﾞの場合
IP,SVRNAME は次のように設定する必要があります。

ﾊﾟﾗﾒｰﾀ	IP ｺﾏﾝﾄﾞ	SVRNAME ｺﾏﾝﾄﾞ
ACTIVE	設定しない	設定する(Passive の IP)
REMOTE	設定する (Remote の IP)	設定しない

③ HSMS-SS で PASSIVE ﾎﾟｰﾄで CLIENT=DEFINE の場合
DISPCH ｺﾏﾝﾄﾞの設定によって、通信ﾛｸﾞのﾁｬﾝﾈﾙ表示だけではなく、PASSIVE 側で、通信に使用する通信ﾊﾟﾗﾒｰﾀもどちらのものを使用するかが変わります。
DISPCH = PASSIVE の場合は、PASSIVE と定義されたﾁｬﾝﾈﾙの通信ﾊﾟﾗﾒｰﾀを使用します。(DVID, ﾌﾟﾛﾄｺﾙﾀｲﾏｰなど) DISPCH = ACTIVE の場合、通信定義ﾌｧｲﾙ内に定義されている IP の REMOTE ACTIVE 装置との通信の場合は、ACTIVE として定義されたﾁｬﾝﾈﾙの通信ﾊﾟﾗﾒｰﾀを使用します。但し、CLIENT ｺﾏﾝﾄﾞが ANY の場合で、定義ﾌｧｲﾙ内に定義されていない IP との通信の場合、PASSIVE ﾁｬﾝﾈﾙの通信ﾊﾟﾗﾒｰﾀが使用されます。

8．2　SECSプロトコル関連コマンド

SECS 通信チャンネルに対するプロトコル関連の設定方法について例を示しながら説明します。

（1）　まず、START コマンドで通信チャンネル番号を指定し、定義開始を宣言します。
CH = 1 の場合は次のように表現します。

```
START = 1                    ; 1ﾁｬﾝﾈﾙの定義を開始する。
```

（2）　次に、SECS 通信であることと、MASTER/SLAVE のどちらであるかをを示す SECS コマンドを指定します。

```
SECS = MASTER                ; MASTER の場合
            または
SECS = SLAVE                 ; SLAVE   “
```

（3）　以下、SECS プロトコル通信に必要なパラメータを指定します。

```
DVID =  2234                 ; ﾃﾞﾊﾞｲｽ ID を設定する(16 進)
PORT =  COM1                 ; ｼﾘｱﾙﾎﾟｰﾄを指定 1 は 1 番目
BAUD =  9600                 ; 通信速度を指定 120,240,2400,4800,9600,14400,19200,  38400
RETRY =3          ; ﾌﾟﾛﾄｺﾙｴﾗｰ検出時のﾘﾄﾗｲ回数
T1    = 10                   ; T1 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 1 秒
T2    = 100                  ; T2 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 10 秒
T3    = 450                  ; T3 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 45 秒
T4    = 450                  ; T4 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 45 秒
```

（4）　デバイス ID チェック,S9Fx 送信指定などを ON/OFF で設定します。
また S9F11 用メッセージ長を設定します。
ON/OFF 指定の場合、ON ならばその機能を実施し、OFF ならば実施しない。

```
S9F0_RSP = ON                ; ON ならばﾌﾞﾛｯｸ番号異常時に S9F0 を送信する。(SECS-I)
S9F1_RSP = ON                ; ON ならばﾃﾞﾊﾞｲｽ IDｴﾗｰ時に S9F1 を送信する。
S9F9_RSP = ON                ; ON ならば T3 ﾀｲﾑｱｳﾄ時に S9F9 を送信する。
S9F11_RSP= OFF               ; ON ならば MAX_S9F11_SIZE を超える長さの MSG 受信時 S9F11 を送信する。
DVID_CK = ON                 ; ON ならばﾃﾞﾊﾞｲｽ ID のﾁｪｯｸを行う。
                           ; ｴﾗｰの場合、S9F1_RSP=ON ならば S9F1 を送信する。
WBLK_CK  = ON                ; ON ならばﾒｯｾｰｼﾞの２重ﾒｯｾｰｼﾞ受信のﾁｪｯｸを行う。
MAX_S9F11_SIZE = 8192        ; ﾁｬﾝﾈﾙが受信できるﾒｯｾｰｼﾞ最大長(ﾊﾞｲﾄ) これを超えると S9F11 を送信
                           ; します。(S9F11_RSP=ON 設定時)
```

（5）　マニュアル通信時で、CH別選択のためのサイド（装置/ホスト）の指定を行います。
(注) ｺﾏﾝﾄﾞはﾊﾞｰｼﾞｮﾝ-3 では無視されます。

```
MANSIDE = EQUIPMENT      ; 装置側の通信を行う。
      または
MANSIDE = HOST           ; ホスト側の通信を行う。
```

（6）　以上の設定が終わったら、１つのチャンネルの設定終了を宣言します。

END

(注)

PORT の COMi の i は、コンピュータに実装されて Windows で云うシリアルポートの番号(1,2,3...)に対応しています。

８．３　ＨＳＭＳプロトコル関連コマンド

HSMS プロトコルチャンネルに対する定義方法について例を示しながら説明します。

（1）　まず、START コマンドで通信チャンネル番号を指定し、定義開始を宣言します。
CH = 3 の場合は次のように表現します。

```
START = 3                    ; 3ﾁｬﾝﾈﾙの定義を開始する。
```

（2）　次に、HSMS 通信であることと、このチャンネルがどのエンティティであるかを示す HSMS コマンドを指定します。

```
HSMS = PASSIVE               ; Passive の場合
HSMS =  ACTIVE               ; Active     “
        または
HSMS =  REMOTE               ; Remote     “
```

（3）　エンティティ別に付属コマンドを指定します。

① PASSIVE の場合

```
CLIENT = ANY                 ; 接続できるｸﾗｲｱﾝﾄ(REMOTE)に制限を付けない。
       または
CLIENT = DEFINE              ; 接続できるｸﾗｲｱﾝﾄはHSMS = REMOTE 定義のもののみ。
```

② ACTIVE の場合

```
SVRNAME = <IP ｱﾄﾞﾚｽ>          ; 接続相手のｻｰﾊﾞｰの IP ｱﾄﾞﾚｽを指定
```

③ REMOTE の場合（ﾘﾓｰﾄの Active）

```
IP      = <IP ｱﾄﾞﾚｽ>         ; 接続相手のｸﾗｲｱﾝﾄの IP ｱﾄﾞﾚｽを指定します。
```

（4）　セッションの種類、 SS(Single Session)か， GS(General Session)かを指定します。

```
SESSION = SS                 ; Single セッションの場合
    または
SESSION = GS                 ; General セッションの場合
```

（5）　HSMS プロトコル通信に必要なパラメータを指定します。

```
DVID =  2234                 ; ｾｯｼｮﾝ ID を設定する(16 進)
PORT =  5000                 ; TCP ﾎﾟｰﾄを指定
T3    = 450                  ; T3 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 45 秒
T5    = 100                  ; T5 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 10 秒
T6    = 50                   ; T6 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 5 秒
T7    = 10                   ; T7 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 10 秒
T8    = 5                    ; T8 ﾌﾟﾛﾄｺﾙﾀｲﾏｰ = 5 秒
LINKTEST = 10                ; LINKTEST.REQ の周期 10 秒
                             ;  値=0 の場合は LINKTEST を行わない。
```

DISPCH=PASSIVE　　；通信ﾛｸﾞ表示するCH番号は=PASSIVEならばPassiveのﾁｬﾝﾈﾙ、
；=ACTIVEならばRemote ActiveのCH番号
通信制御ﾊﾟﾗﾒｰﾀも選択されたﾁｬﾝﾈﾙのものが使用される。

（6）　デバイスID,S9Fx送信指定などをON/OFFで設定します。またS9F11用メッセージ長を設定します。
ON/OFF指定の場合、ONならばその機能を実施し、OFFならば実施しない。

S9F1_RSP = ON　　；ONならばﾃﾞﾊﾞｲｽIDｴﾗｰ時にS9F1を送信する。
S9F9_RSP = ON　　；ONならばT3ﾀｲﾑｱｳﾄ時にS9F9を送信する。
S9F11_RSP= OFF　　；ONならばMAX_S9F11_SIZEを超える長さのMSG受信時S9F11を送信する。
DVID_CK　= ON　　；ﾃONならばﾊﾞｲｽIDのﾁｪｯｸを行う。
；ｴﾗｰの場合、S9F1_RSP=ONならばS9F1を送信する。
WBLK_CK　= ON　　；ONならばﾒｯｾｰｼﾞの２重ﾒｯｾｰｼﾞ受信のﾁｪｯｸを行う。
MAX_S9F11_SIZE = 8192　　；ﾁｬﾝﾈﾙが受信できるﾒｯｾｰｼﾞ最大長(ﾊﾞｲﾄ)これを超えるとS9F11を送信
；します。(S9F11_RSP=ON設定時)

（7）　マニュアル通信時で、ＣＨ別選択のためのサイド（装置/ホスト）の指定を行います。
(注) ｺﾏﾝﾄﾞはﾊﾞｰｼﾞｮﾝ-3では無視されます。

MANSIDE = EQUIPMENT　　；装置側の通信を行う。
　　または
MANSIDE = HOST　　；ホスト側の通信を行う。

（8）　以上の設定が終わったら、１つのチャンネルの設定終了を宣言します。

END

(注)
HSMSの場合、PASSIVEとして通信を開始した場合、Active側からの接続を待ちますが、受付けることができるのは、CLIENTコマンドの設定によって変わります。
①　CLIENT = ANYの場合
通信環境定義ファイルに定義されていないIPのACTIVE相手からの接続も受け付けます。
②　CLIENT = DEFINEの場合
通信環境定義ファイルの中でIP コマンドでIPｱﾄﾞﾚｽが設定されているチャンネルの相手装置からの接続のみを受け付けます。

先ほども書きましたが、IP,SVRNAMEｺﾏﾝﾄﾞの指定とチャンネルの関係はつぎのようになります。

Entity	IPｺﾏﾝﾄﾞ	SVRNAME
PASSIVE	指定しない	指定しない
ACTIVE	指定しない	指定する
PASSIVE/ACTIVE	指定する	指定しない

８．４　その他の定義コマンド

その他のコマンドについて説明します。

（１）　LOGFILE コマンド

LOGFILE　=〈処理履歴ファイル名〉

処理履歴のログファイル名を指定します。
例 LOGFILE = C:¥SECS¥LOG¥SIMLOG.LOG

本コマンドの指定がない場合は、シミュレータと同じディスクフォルダー内に PRLOG.LOG の名前のファイルにログが記録されます。

（２）　LOGFMT コマンド

LOGFMT　=〈FMT〉

本コマンドは、SECS 通信メッセージのログ形式を設定します。
〈FMT〉は、SECS メッセージのログ形式を指定します。〈FMT〉=LIST、ITEM、HEAD のどちらかを指定します。
(例 LOGFMT = ITEM)
LIST の場合はリスト構造表現になり、ITEM の場合はアイテムの羅列表現になり、の場合、ヘダー表示だけになります。

本コマンドの設定がない場合は、デフォルトで LIST 構造形式になります。

（３）　MSNAMEi コマンド

(注) ｺﾏﾝﾄﾞはﾊﾞｰｼﾞｮﾝ-3 では無視されます。

MSNAMEi　=〈MAILSLOTNAME〉
　　i = １～8
本コマンドは、シミュレータが他のアプリケーションプロセスプログラムとプロセス間通信を行うためにメールスロットを使用することができますが、そのメールスロットの名前を指定します。

本コマンドで設定したメールスロット名はシーケンスプログラムの中で MSNAME コマンドで取出すことができます。

例　MSNAME1 = ¥¥.¥mailslot¥proA_to_sim
　　MSNAME2 = ¥¥.¥mailslot¥sim_to_proA

（４）　HINT コマンド

HINT = 〈ON/OFF〉

シミュレータ画面で操作ヒント表示を行うかどうかを指定します。
ON ならば表示をします。 OFF ならば表示をしません。
本コマンドが指定されない場合、デフォルトとして ON 状態になります。
例　HINT = OFF

（５） MANCH コマンド

(注) ｺﾏﾝﾄﾞはﾊﾞｰｼﾞｮﾝ-3 では無視されます。

MANCH = CHNO, CHNO,...CHNO
マニュアル通信開始時にどのチャンネルを実行開始するかを指定することができます。
本コマンドで指定されたチャンネルは、マニュアル通信実行開始時のチャンネル選択画面で、チェックされた状態で表示されますので、マウスのクリック操作を省略することができます。
MANCH = 1,2,3,4,5,6,7,8 と指定すれば,全チャンネルを選択できます。

（６） MANSIDE コマンド

(注) ｺﾏﾝﾄﾞはﾊﾞｰｼﾞｮﾝ-3 では無視されます。

MANSIDE = HOST または EQUIPMENT

マニュアル通信開始時に全ＣＨ選択が選択されたときのシミュレータ全体のサイドを指定します。

(注)
START と END コマンドの間で使用される MANSIDE コマンドは、START コマンドで指定されたチャンネルだけのサイドを指定します。
START,END の間以外で定義された場合は、全ＣＨ選択モードのためのサイドになります。

（７） MAX_LENGTH コマンド

MAX_LENGTH = <サイズ>

シミュレータが受信できるメッセージの最大長をバイト単位で設定する。
デフォルト値は 65536 バイトで、設定できる最大値は 256K バイトである。
HSMS プロトコルの場合、この値を超える長さのメッセージが到達したときは接続を切断し、再接続を行う。

（８） CH_LOG コマンド (2006.3 新規追加コマンド)

CH_LOG = <ON/OFF>

通信チャンネル別に通信ログをファイルに記録するかどうかを指定します。
ON の場合は、送受信した通信メッセージをチャンネル別のログファイルに記録します。
ログファイル名は以下のように予約されています。
ch?_log.log － ? 部分がチャンネル番号になります。
OFF の場合は、チャンネル別の通信ログファイルは記録されません。

９．　ＳＥＣＳ-II通信メッセージ定義ファイル仕様

ＳＥＣＳ／ＨＳＨＳシミュレータを動作させるための準備作業の中で、まず第１に必要なことは、ホスト／装置間でやり取りされるＳＥＣＳ-II レベルのメッセージ定義仕様に基づいて、それらをメッセージ定義ファイル上に書き表すことになります。

本シミュレータが採用するメッセージ定義形式として次の２つのものがあります。

（1）　ＭＳＧ形式　　：　本シミュレータ専用形式です。（こちらの形式を推奨します。）
（2）　ＳＭＬＸ形式　：　ＳＭＬ形式を本シミュレータ向けにした形式です。

何れの形式もメッセージをリスト構造で表現します。

メッセージとそれに含まれるデータ数について下記の制限事項があります。

番号	項　目	制限内容
1.	登録ﾒｯｾｰｼﾞ数	最大 1024 個（定義ﾌｧｲﾙあたり）
2.	ﾒｯｾｰｼﾞ長 （送受信とも）	ﾍﾀﾞｰを含めて最大 256K ﾊﾞｲﾄ　ﾃﾞﾌｫﾙﾄ=65536 ﾊﾞｲﾄ (環境定義ﾌｧｲﾙの設定に基づく)
3.	ﾒｯｾｰｼﾞに含まれるｱｲﾃﾑ数	最大 8192 個(LIST ｱｲﾃﾑを除く)
4.	ｱｲﾃﾑに含まれるﾃﾞｰﾀ数	A,J　　：最大 8192 ﾊﾞｲﾄ その他 ：最大 8192 個

９．１　ＭＳＧ形式のメッセージ定義

（1）　基本形式は次のとおりです。

```
[MSGNAME]　[MSGID]　[W_BIT] [SIDE] [ITEM DATA 群] [TERMINATOR]
```

MSGNAME

シミュレータが識別できるメッセージ名であり、メッセージファイル内でユニークでなければなりません。使用できる文字は、全て半角文字であり、全角文字は使用できません。
また、頭文字が英文字であり、残りは全て英数字またはアンダーバーから成る文字列で、長さは 20 文字以内でなければなりません。

MSGID

SxFy のように表現します。ここで、x がストリーム、 y がファンクションコードになります。

W_BIT

１次メッセージで、２次応答メッセージを期待するメッセージの場合、’W’　と表現します。それ以外のメッセージ(2 次メッセージを期待しない、または２次メッセージ)には　‘W’　を付けないでください。

SIDE

ホスト／装置側のどちら側が送信するメッセージなのかを識別します。装置側の場合、’E’　を付け、ホストの場合は何も付けません。

ITEM DATA 群

アイテムを‘＜＞’で囲み、‘＜＞’の中にアイテム記号、バイトサイズ、アイテム名、値を次のように記述します。

LIST は ‘<L’ で表現し、 その後、LIST に含まれるデータアイテムを ‘>’ の中に含めます。LIST の中に LIST を入れ子にすることもできます。

LIST 以外のアイテムは、’< ITEM 記号[n] ITEM 名=値 >’の形式で記述します。
ｎはバイト長であり、複数個の値を表現する場合、値と値の間を ’,’ (ｶﾝﾏ) で区切ります。

配列表現として、固定長と可変長があります。
固定長 ： [n] のように表現します。
可変長 ： [min..max] のように表現します。
min ： 最小ﾊﾞｲﾄ長 (短くても min ﾊﾞｲﾄのﾃﾞｰﾀがｾｯﾄされる。)
max ： 最大ﾊﾞｲﾄ長

文字列の場合、文字列長を size とすると、size < n または、 size < min の場合、サイズ分に満たない後ろの部分に空白(Space)文字が詰められます。
数値データの場合、値が指定されていないものにはゼロ(0)が設定されます。

TERMINATOR

メッセージの終端を ‘¥’ 文字で明示します。

（2） データアイテム値の表現

ITEM 記号	ITEM 名	単位ｻｲｽﾞ	表現方法	例
B	Binary	1 (byte)	(1) 10 進表現の場合、そのまま数値で表現します。 (2) 16 進表現の場合、 頭に’X’または 0x を付けて 16 進ﾃﾞｰﾀ値を続けます。	23 X12 0x12
T	Boolean	1	(1) 10 進表現の場合、そのまま数値で記述します。 (2) 論理表現の場合、’T’または’F’で記述します。T=真, F=偽 となります。	1 T, F
A J	ASCII	1	設定したい文字列を ” (二重引用符) で囲んで記述します。 配列ｻｲｽﾞに満たない場合は、残りの部分には空白(Space)ｺｰﾄﾞが詰められます。	“TEST STRING”
S1 or I1 S2 or I2 S4 or I4 S8 or I8	Integer	1 2 4 8	(1) 10 進または 16 進で記述できます。 10 進の場合､符号+,-を付けることができますが、16 進の場合、符号を付けることはできません。	17, -18 X11, XF8 0x11, 0xf8
U1 U2 U4 U8	Unsigned Integer	1 2 4 8	(1) 10 進または 16 進で記述できます。 (2) U8 の場合,32 ﾋﾞｯﾄを超える値については 16 進でしか表現できません。	23, 45 X16, X2D 0x16,0x2d
F D	4 Bytes 8 Bytes Floating Data	4 8	実数、指数のどちらかで表現します。	1.23, -3.4 1.23E-01

（3） 特殊アイテム名

文字列タイプのデータアイテム名が、“@TIME”として名付けられた場合、メッセージ送信時にシミュレータは@TIME に含まれる文字列データの内容によって特殊処理を行います。

データアイテムを次のように記述します。

〈 A[16] @TIME = “YYYYMMDDHHNNSSCC” 〉

ここで、YYYY ： 西暦の年を表します。最大４文字です。４文字より少ない場合は、年データの下位桁から指定文字数だけ編集し、設定します。

MM ： 月、 DD ： 日、 HH ： 時、 NN ： 分、 SS ： 秒、 CC: 1/100 秒 を表します。編集したくないデータは省略することができます。

例えば、月日時分だけを編集したい場合“MMDDHHNN”のようにします。

（4） 特殊文字

A,J アイテムの中に、表示できない制御コードを送信したい場合は、次のように表現します。

① エスケープ文字 ¥の後ろに制御コードに対応する特定の文字をつけます。

② ¥¥ は ¥とみなします。

③ ¥” は “ とみなします。(2 重引用符)

④ コード 01～1F 文字コードは ¥ + [文字](文字は A～Z,[,|,],^,_) として表現します。

[文字] 8 ビットのコードの上位３ビットを０とし、下位５ビットをマスクした値とします。

コード生成早見表を示します。

制御ｺｰﾄﾞ(16 進)	表 現	制御ｺｰﾄﾞ(16 進)	表 現
00	¥@ (*1)	10	¥P
01	¥A	11	¥Q
02	¥B	12	¥R
03	¥C	13	¥S
04	¥D	14	¥T
05	¥E	15	¥U
06	¥F	16	¥V
07	¥G	17	¥W
08	¥H	18	¥X
09	¥I	19	¥Y
0A	¥J	1A	¥X
0B	¥K	1B	¥[
0C	¥L	1C	¥| (*2)
0D	¥M	1D	¥]
0E	¥N	1E	¥^
0F	¥O	1F	¥_

(*1) コード 00 は文字列の終端を示す文字コードです。

従って、この文字コードが文字列の途中に含まれている場合、このコード以降は通常表示されないことになります。

(*2) ¥¥ は ¥文字とみなされますので、コード=0C のためには, | 文字を使用してください。

（5） コメントの付加

A,J の文字列定義内（”で囲まれている内部）を除く、行の先頭あるいは、途中に ‘;’(ｾﾐｺﾛﾝ) を検出した場合、; 以降をコメントとみなし、メッセージ定義に関係ないものとみなし無視します。

（6） メッセージのチャンネル指定

（注） 本指定は、バージョン－３．０では、無視されます。

メッセージ定義行の前の行で、$CH コマンドを使って、その後に定義されるメッセージは$CH で指定されたチャンネル用に予約することができます。

表現は $CH = <CHi, CHj, ...>¥ と表現します。(最後に"¥"をつけて下さい。)

メッセージ定義ファイル内に$CH コマンドの指定が全くない場合、全チャンネルが全メッセージを使用できることになります。

このチャンネル予約メッセージの機能は、マニュアル通信時で、チャンネル単位での送信モードのときだけ有効になります。

（7） メッセージ例

```
S1F13_H S1F13 W
     <L
     >¥
S1F14_E S1F14 E
     <L
          <B[1]COMMACK = x00>
          <L
              <A[6]MDLN = "xxxxxx">
              <A[6]SOFTREV = "VER1.0">
          >
     >¥

S1F13_E S1F13 W E
     <L
          <A[6]MDLN = "xxxxxx">
          <A[6]SOFTREV = "VER1.0">
     >¥
S1F14_H S1F14
     <L
          <B[1]COMMACK = x00>
         <L
         >
     >¥

S5F1 E W S5F1                    ; msg_name, 装置ｻｲﾄﾞ,Wbit=1, S5F1
      <L
          <B[1] ALCD = 1>
          <U4[4] ALID = X123>
          <A[40] ALTX = “ALARM REPORT-1” >
     >¥
```

９．２　ＳＭＬＸ形式のメッセージ定義

元々のＳＭＬ形式に対し、本シミュレータの機能に必要な要素を追加変更し、編集した形式を**ＳＭＬＸ**形式と呼ぶことにします。

追加要素は①HOST/装置のサイド、②ｱｲﾃﾑ名であり、変更要素として、③ﾒｯｾｰｼﾞ名の表現法があります。

プロジェクト生成プログラムは、ＳＭＬ形式のメッセージファイルをＳＭＬＸ形式に変換する機能を提供しています。本機能を使えば、簡単な操作で①ｻｲﾄﾞ、②各ﾃﾞｰﾀｱｲﾃﾑに名前を付加したメッセージファイルを作成することができます。

（1）　SMLX の基本形式は次のとおりです。

[MSGNAME]　[MSGID]　[W_BIT] [SIDE] [ITEM DATA 群] [TERMINATOR]

MSGNAME

シミュレータが識別できるメッセージ名であり、メッセージファイル内でユニークでなければなりません。使用できる文字は、全て半角文字であり、全角文字は使用できません。
また、頭文字が英文字であり、残りは全て英数字またはアンダーバーから成る文字列で、長さは 20 文字以内でなければなりません。

MSGID

SxFy のように表現する。ここで、x がストリーム、 y がファンクションコードになります。

W_BIT

１次メッセージで、２次応答メッセージを期待するメッセージの場合、'W' と表現します。それ以外のメッセージは 'W' を付けてはいけません。

SIDE

ホスト／装置側のどちら側が送信するメッセージなのかを識別します。
装置側の場合、'E' を付け、ホストの場合は何も付けません。

ITEM DATA 群

アイテムを '< >' で囲み、 '< >' の中にアイテム記号、バイトサイズ、アイテム名、値を次のように記述します。
LIST は 'L' で表現し、 その後、LIST に含まれるデータアイテムを '>' の中に含めます。LIST の中に LIST を入れ子にすることもできます。

LIST 以外のアイテムは、'< ITEM名[n] ITEM名 値 >'の形式で表現します。
（MSG形式と違って ITEM名と値との間に '=' が無い。）

配列の中に複数個の値を表現する場合、値と値の間を ' ' (Space)で区切ります。

配列の表現はバイト数ではなく、データ数になります。MSG形式と同様に、固定長と可変長の表現ができます。
固定長　：　[n] のように表現します。
可変長　：　[min..max] のように表現します。
min ： 最小ﾃﾞｰﾀ数（短くても min のﾃﾞｰﾀ数がｾｯﾄされる。）
max ： 最大ﾃﾞｰﾀ数

但し、SMLX では、配列の表現[n] の表現を省略することも可能です。この場合、単に、データ値を並べることになります。

例 〈A TEXT “1234567890”〉　　〈 B DBIN 0x12 0x23 0x34〉

文字列の場合、文字列長を size とすると、size 〈 n または、 size 〈 min の場合、size に満たない後ろの部分に空白(Space)文字が詰められます。

TERMINATOR　　：メッセージの終端を ‘.’ 文字で明示します。

（2） データアイテム値の表現

ITEM 記号	ITEM 名	表現方法	例
B	Binary	(1) 10 進表現の場合、そのまま数値で表現します。 (2) 16 進表現の場合、 頭に’0x’を付けて、16 進ﾃﾞｰﾀ値を続けます。	23 0x12
BOOLEAN	Boolean	(1) 10 進または 16 進表現で定義します。	1
A J	ASCII	設定したい文字列を’”’ (二重引用符) で囲んで表現します。 配列ｻｲｽﾞに満たない場合は、残りの部分には空白(Space)ｺｰﾄﾞが詰められます。	“TEST STRING”
I1 I2 I4 I8	Integer	(1) 10 進または 16 進で表現できます。10 進の場合、符号+,-を付けることができますが、16 進の場合、符号を付けることはできません。 (2) S8 の場合,32 ﾋﾞｯﾄを超える値については 16 進でしか表現できません。	17, -18 0x11, 0xF8
U1 U2 U4 U8	Unsigned Integer	(1) 10 進または 16 進で表現できます。 (2) U8 の場合,32 ﾋﾞｯﾄを超える値については 16 進でしか表現できません。	23, 45 0x16, 0x2D
F4 F8	4,8 Byte Floating Data	実数、指数のどちらかで表現します。	1.23, -3.4 1.23E-01

（3） 特殊アイテム名

MSG 形式の場合と全く同様に、日付時刻データの表現ができます。

（4） 特殊文字

A,J タイプのアイテムのケースで、文字列の中に表示できない制御コードを含ませたい場合は、16 進数値で表現します。16 進表現では数値の頭に、”0x”を付けます。
例えば、改行コード LF (10 進コード=10)は、 0xA のように表現します。

特殊コードと通常の文字列を次のように合成することができます。
〈 A ASTR 0x01 “1111 2222” 0x0A 〉

これは、次のような16進文字コードのデータ列を作ります。

01 31 31 31 31 20 32 32 32 32 0A

（5） コメントの付加

A,J の文字列定義内(<>内) を除く、行の先頭あるいは、途中に ‘//’を検出した場合、// 以降をコメントとみなし、メッセージ定義には関係ないものとみなし無視します。

また、’/*’ で始まり、 ‘*/’ で終わる部分はコメントとみなされます。この場合、コメントは複数行に跨っても構いません。

（6） メッセージ例

```
/* S5F1 アラーム報告　その１ */
S5F1 E W S5F1
       <L
           <B[1] ALCD = 1>
           <U4[1] ALID = 0x123>
           <A[40] ALTX = “ALARM REPORT-1”>
       >.
```

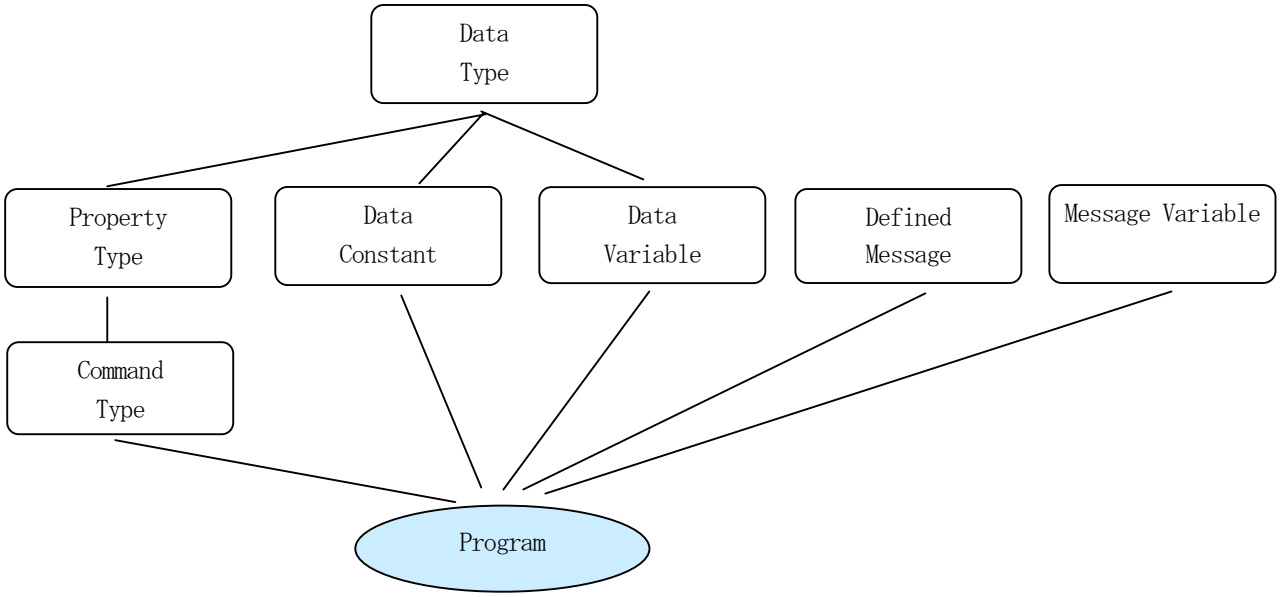
10．プログラム関連基本情報と定義ファイル

本シミュレータバージョン３．０が有するプログラム関連基本情報の定義仕様と定義ファイルについて説明します。

プログラム関連定義情報として、以下の情報があります。

No.	関連定義情報	内容	定義ﾌｧｲﾙ名
1.	基本ﾃﾞｰﾀﾀｲﾌﾟ	本ｼﾐｭﾚｰﾀが定義する基本ﾃﾞｰﾀﾀｲﾌﾟ	（なし）
2.	ﾌﾟﾛﾊﾟﾃｨﾀｲﾌﾟ	基本ﾃﾞｰﾀに基づくﾌﾟﾛﾊﾟﾃｨ情報	property.txt
3.	定数ﾃﾞｰﾀ情報	定数ﾃﾞｰﾀ情報	constant.txt
4.	ﾃﾞｰﾀ変数情報	数値、文字列ﾃﾞｰﾀ変数情報	variable.txt
5.	ﾒｯｾｰｼﾞ変数情報	ﾒｯｾｰｼﾞ変数	msgvar.txt
6.	ｺﾏﾝﾄﾞﾀｲﾌﾟ情報	ﾌﾟﾛｸﾞﾗﾑｺﾏﾝﾄﾞ	command.txt
7.	定義ﾒｯｾｰｼﾞ情報	SECS-II 通信ﾒｯｾｰｼﾞ（ﾕｰｻﾞが定義する）	ﾕｰｻﾞ作成による

それぞれの情報の関係は次のようになります。



10．1　基本データタイプ

データタイプとデータ表現についてを参照ください。

10．2　プロパティタイプ

データタイプに基づくプログラムコマンドのプロパティとして使用される情報です。
プロパティ・タイプ定義は次のように記述します。

（1）書式

P_XXXX = <基本ﾃﾞｰﾀﾀｲﾌﾟ>{
　CAPTION:　<“ﾌﾟﾛﾊﾟﾃｨ表示名”>
　RANGE:　<値の範囲またはとりうる値>
　DEFAULT:　<値>

CAP0:　　〈”値-01 の表示名”〉
CAP1:　　〈”値-1 の表示名”〉
.
.
CAPn:　　〈”値-n の表示名”〉
OPT:　　〈ｵﾌﾟｼｮﾝの指定〉
}

（2）説明

① **P_XXXX** は定義したいプロパティ名である。
'=' の後の**〈基本ﾃﾞｰﾀﾀｲﾌﾟ〉** でとり得るデータ・タイプを指定する。
'{' 以降、**'}'** までが定義内容である。 各プロパティ要素の項目名には **':'** 文字を付加する。

② **CAPTION** は、画面上で表示するための当該プロパティのデフォルト名。
〈”ﾌﾟﾛﾊﾟﾃｨ表示名”〉で与える。プロパティ表示名は必ず、二重引用符(”)で囲むこと。

③ **RANGE** は、そのプロパティがとりうる値の範囲またはとり得る値（複数）を指定する。
範囲の場合は、**dmin..dmax** のように値と値の間を **'..'** でつないで表現し、最小値が dmin で最大値が dmax の範囲の値をとりうることを意味する。
とりうる値が限定されている（連続あるいは飛び飛びに）ものについては、d1,d2,d3,..dn4 のように ',' 切りでとりうる値を列挙して表現する。

④ **DEFAULT** は初期値を指定する。

⑤ **CAP0～CAPn** は RANGE で与えられた min, min+1, など。
、とりうる値に対する表示名を指定するために使用する。
この項目は、整数の値をとるデータ・タイプについてのみ適用される。
CAPi の i は、０から始まり、RANGE で与えられた i 番目の値に対応する。
例えば、BOOLEAN の場合、 値が 0..1 と与えられるが、CAP0 は 0(FALSE)に対し、CAP1 は 1(TRUE)に対する表示名になる。

⑥ 行の最初の**'}' 文字**は定義の終わりを示す。

⑦ 上記□～□までの各項目はそれぞれ１行で表現しなければならない。

⑧ 各行の改行前に 現れる'**//**'（/文字２個の並び）以降は、行の終わりまで、コメントとみなされる。

⑨ 各プロパティ・タイプの間または、各項目行の間に空行またはコメントだけの行を含めてもよい。

プロパティ・タイプ名に使用される文字は半角英数字と '_'(ｱﾝﾀﾞｰｽｺｱ)とする。英字については大文字、小文字を問わない。(内部処理はすべて大文字に変換して処理する。)

（3）定義例

① 32 ビット整数データ

```
P_INT32 = INT32{
        caption:    "32ﾋﾞｯﾄ整数/変数"
        range:      -2147483648..2147483647
        default:    0
        }
```

② ファイルアクセスモードのプロパティ

```
P_FILE_MODE = INT8{
        caption:    "ﾌｧｲﾙｱｸｾｽﾓｰﾄﾞ"
        range:      0..5
        default:    2
        cap0:       "r"
        cap1:       "w"
        cap2:       "a"
```

```
cap3:       "r+"
cap4:       "w+"
cap5:       "a+"
}
```

10．3　定数情報

ユーザが作成するシミュレータプログラムが使用する定数です。次のように定義します。

（1）書式

```
K_XXXX = <基本ﾃﾞｰﾀﾀｲﾌﾟ>{
   CAPTION:  <“定数表示名”>
   VALUE:    <値>
}
```

（2）説明

□ **K_XXXX** は定義したい定数名である。

’=’ の後の**<基本ﾃﾞｰﾀﾀｲﾌﾟ>** でとり得るデータ・タイプを指定する。

‘{‘ 以降、’}’ までが定義内容である。

② **CAPTION** は、画面上で表示するための名前

<”定数表示名”>で与える。必ず二重引用符(”)で囲むこと。

③ **VALUE** は、定数の値である。16 進数で表現する場合は頭に “0x” を付ける。

（3）定義例

① プロジェクトの最大 PU 数

```
K_MAX_PU  = INT32{
        caption:    "最大ﾌﾟﾛｸﾞﾗﾑﾕﾆｯﾄ数"
        value:      15
        }
```

10．4　データ変数情報

シミュレータが使用するデータ変数です。プログラムで使用することができます。
変数には、アクセスができるプログラム空間によって（2）-①で説明する３種類の変数があります。
次のように定義します。

（1）書式

```
VXXXX = <基本ﾃﾞｰﾀﾀｲﾌﾟ>{
   CAPTION:  <“変数表示名”>
   VALUE:    <値>
}
```

（2）説明

① **VXXXX** は定義したい変数名である。

’=’ の後の**<基本ﾃﾞｰﾀﾀｲﾌﾟ>** でとり得るデータ・タイプを指定する。

‘{‘ 以降、’}’ までが定義内容である。 各プロパティ要素の項目名には ‘:’ 文字を付加する。

変数名の頭文字によって、変数の属性が決まる。

文字　：　**システム変数**を意味する。全プロジェクトが共通で使用できる。

@　　：　プロジェクトの **PU 内ローカル変数**

#,@以外 ：　プロジェクト内**グローバル変数、**

② CAPTION は、画面上で表示するための名前
　　<”変数表示名”>で与える。必ず二重引用符(”)で囲むこと。
③ VALUE は、変数の初期値である。16 進数で表現する場合は頭に “0x” を付ける。

（3）定義例
① SECS ITEM A のアイテムコード(16 進) システム変数
```
#I_A = P_INT32{
        caption:    "ASCII"
        size:       1
        value:      0x40
  }
```
② 通信チャンネル番号　プロジェクト内グローバル内変数
```
chno = P_INT32{
        caption:    "通信ﾁｬﾝﾈﾙ番号"
        size:       1
        value:      1
  }
```
③ 通信チャンネル番号　PU 内ローカル変数
```
@chno = P_INT32{
        caption:    "通信ﾁｬﾝﾈﾙ番号"
        size:       1
        value:      1
  }
```

10．5　メッセージ変数

送受信するときに使用するメッセージを格納するための変数です。定義メッセージからの代入、相手装置からの受信時に実体を持つことになります。

（1）書式
MXXXX

（2）説明
単に変数名だけを記述します。
① MXXXX は定義したいメッセージ変数名である。
　　@　　：　プロジェクトの PU 内ローカル変数
　　@以外　：　プロジェクト内グローバル変数、

（3）定義例
① グローバル
　　SMSG
② ローカル
　　@RMSG

10．6　プログラムコマンドタイプ情報

DSH シミュレータのプログラムを組むためのコマンドです。プロジェクト、プログラムユニット制御に使用するコマンドから通信制御など全てのコマンドを以下の仕様で定義します。
なお、カスタム・デザインのコマンドも定義し、組み込むことも可能です。

（1）書式

```
CCCCC = COMMAND{
    PROPERTY(
        CAPTION:  〈"ｺﾏﾝﾄﾞ・ｸﾗｽ名"〉
        NAME:     〈"名称"〉
        PARA1:    〈ﾌﾟﾛﾊﾟﾃｨ・ﾀｲﾌﾟ〉,  〈"表示名称"〉, 〈ﾃﾞﾌｫﾙﾄ値/変数名〉, 〈編集ﾀｲﾌﾟ〉
        PARA2:    〈ﾌﾟﾛﾊﾟﾃｨ・ﾀｲﾌﾟ〉,  〈"表示名称"〉, 〈ﾃﾞﾌｫﾙﾄ値/変数名〉, 〈編集ﾀｲﾌﾟ〉
         .
        PARAn:    〈ﾌﾟﾛﾊﾟﾃｨ・ﾀｲﾌﾟ〉,  〈"表示名称"〉, 〈ﾃﾞﾌｫﾙﾄ値/変数名〉, 〈編集ﾀｲﾌﾟ〉
    )
    METHOD(
        FUNCTION:〈対応 API 関数名〉
    )
}
```

（2）説明

① CCCCC は定義したいコマンドである。
　’=’ のあと、COMMAND と続け、定義がコマンド・クラスに対するものであることを表す。
　CCCCC から ‘}’までの内容は書式に書かれているとおり、各行はすべて別々の行に書かれていなければならない。

② ’{‘ 以降 ‘}’までが定義内容である。

③ 定義部分は、PROPERTY 部と METHOD 部の２つに区分される。
　各部の内容は ’（ ‘ と ’）’ で囲まれる。各部内には１個以上の要素が定義される。

④ ROPERTY 部
　PROPERTY 部内にプリミティブが有する全プロパティを定義する。

⑤ METHOD 部
　METHOD 部は、当該プリミティブがＫ－ＢＯＸの実行系で実行される際に、プリミテイブ機能の処理を行うための API 関数名とその関数の引数名ならびに結果格納変数名を指定する。
　（注）この API 関数名は実際の処理実行 API 関数名ではなく、間に入るラッピング関数名。

（3）定義例

```
DEF_PU  = COMMAND{
        property(
            CAPTION:    "ﾌﾟﾛｸﾞﾗﾑ・ﾕﾆｯﾄ定義"
            PU_NAME:    P_PU_NAME,"PU 名","main",M_WIN_PU_NAME,1
            PU_CAPT:    P_COMMENT,"表示名","main",M_WIN_STR_LIT
            ERR_VAR:    P_INT32, "ｴﾗｰ変数", @error,M_WIN_INT_VAR,1
        )
        method(
            FUNCTION:   SIM_def_pu
        )
}
```

11．通信ログファイルから通信メッセージファイルを作成

DSHSIM-V3.0シミュレータで得られた通信ログファイルに記録されているSECS-IIメッセージログからDSHSIM-V3.0で使用できる通信メッセージ定義ファイルを作成することができます。
通信ログファイルは、デフォルトで PROLOG.LOG の名前のファイルです。別の名前で保存されたファイルでもかまいません。
作成されるメッセージ定義ファイルは、拡張子 “.MSG”を持つファイルです。

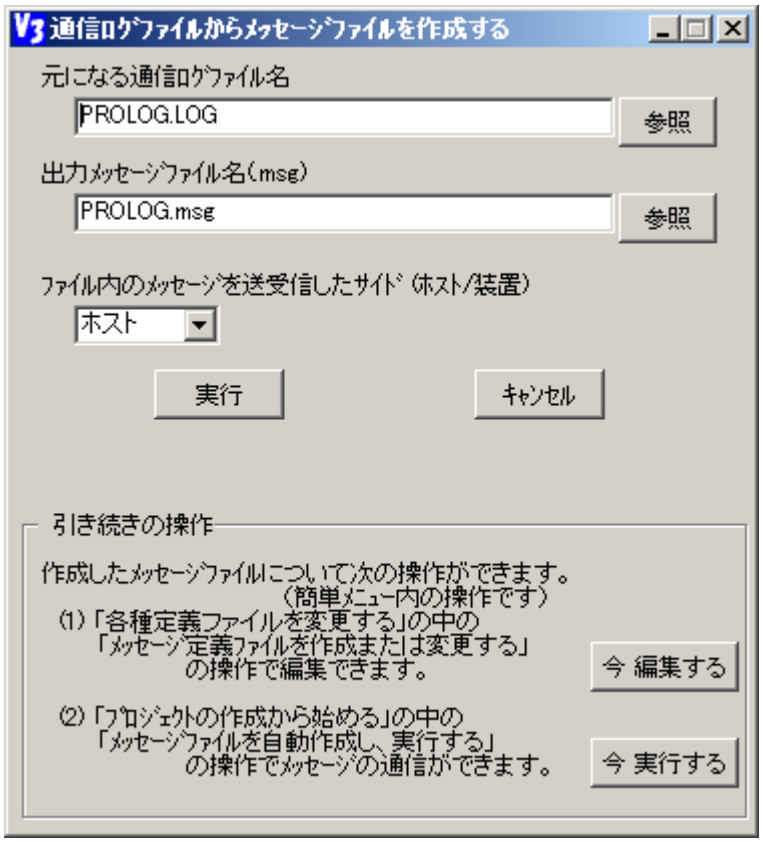
ここで説明する機能によって、指定されたログファイルに含まれる全てのSECS-II メッセージ（S9Fxx を除きます）を含むメッセージ定義ファイルを簡単に作成することができます。

この機能は、相手装置が有する構造が未明のメッセージを実際に接続し送受信を行い、そこで得られた通信ログファイルに含まれる受信メッセージをDSHSIM　V-3.0 で使用できるメッセージファイルに変換し、再利用したいときに便利な機能です。

（注）通信ログを取る際、ファイルPROLOG.LOG を一旦、空にした上で取りたい場合は、処理状況画面の ﾌｧｲﾙ(F) メニュー内の 現処理履歴ﾌｧｲﾙを保存する (S) タブのクリックによる操作でPROLOG.LOGを空にしてください。

（1）操作画面

操作画面は、簡単メニューの ログファイルから通信メッセージ定義ファイルを作る の選択と開始、またはメイン画面のツール(T) ログファイルから通信定義ファイルを作成(M) のクリックで表示させることができます。



（2）操作手順

操作は簡単です。以下のように行ないます。

①最初に、画面の元になる通信ﾛｸﾞﾌｧｲﾙ名の欄にDSHSIMで記録された通信ログファイル名を入力します。
参照ボタンを使って、ファイルを開くダイアログ画面を使って選択することもできます。

②次に、出力したいﾒｯｾｰｼﾞﾌｧｲﾙ名の欄に保存したいメッセージ定義ファイル名を入力します。
参照ボタンを使って、ファイルを保存するダイアログ画面を使って指定することもできます。

③入力ファイルになるログファイルが、ホスト／装置のどちら側として送受信したのかを選択します。

④最後に実行ボタンをクリックすれば作成処理が行われます。そして、実行ボタンの下に結果が表示されます。
成功すると、例えば、上の画面では、“OK: PROLOG.msg Created”と表示されます。
元に指定したファイルが見つからなければ、ファイルが見つかりませんのメッセージ画面がポップアップされます。

操作は、ｷｬﾝｾﾙボタンのクリックで終了することができます。

（3）引き続きの操作

メッセージファイルの作成の後、メッセージファイルの編集あるいは、プロジェクトファイルの生成と実行の操作に進むことができます。

今 編集する ボタンのクリックによって、メッセージファイル編集の操作に進むことができます。

今 実行する ボタンのクリックによって、プロジェクトの生成と実行操作にすすむことができます。

（4）作成例

［ログファイルの内容］ - PROLOG.LOG

次ページに作成されたメッセージファイルの内容を示します。

```
06.01 08:49:11 ppj->lvar_id_tab[PU_SEQ] = 0
06.01 08:49:11 --> prj=DSHENG3 pu=pu_recv start of processing
06.01 08:49:12 PU_SEQ=31 MAX_PU=64
06.01 08:49:13      CH-1 failed in connection with 192.168.1.2
06.01 08:49:24 CH-1 送 dvid=ffff P=0 S=1 (Select.Req)    len=0010 sybt=00000001
06.01 08:49:25 CH-1 受 dvid=ffff P=0 S=2 (Select.Rsp  status=0) len=0010 sybt=00000001
06.01 08:49:49 CH-1 送 S6F11   len=0272 dvid=1234  W  blk=0000 sybt=00000004
               <L 3
                 <U4[4]=2>
                 <U4[4]=999>
                 <L 1
                   <L 2
                     <U2[2]=3>
                     <L 14
                       <B[4]=x0a,x14,x1e,x28>
                       <T[4]=T,F,F,T>
                       <S1[4]=10,20,-30,-40>
                       <U1[4]=50,60,70,80>
                       <S2[8]=100,200,-300,-400>
                       <U2[8]=500,600,700,800>
                       <S4[16]=1000,2000,-3000,-4000>
                       <U4[16]=5000,6000,7000,8000>
                       <S8[32]=x0000000000002710,x0000000000004e20,xffffffffffff8ad0,xffffffffffff63c0>
                       <U8[32]=x000000000000c350,x000000000000ea60,x0000000000011170,x0000000000013880>
                       <E[16]=1.200000,2.300000,-3.400000,-5.500000>
                       <D[32]=1.230000,2.340000,-3.450000,-5.560000>
                       <A[17]="A1234567890ABCDEF">
                       <J[17]="J1234567890ABCDEF">
                     >
                   >
                 >
               >
06.01 08:49:50 CH-1 受 S6F12   len=0013 dvid=1234     blk=0000 sybt=00000004
               <B[1]=x00>
```

[作成されたメッセージファイル] - PROLOG.MSG

```
;---- DSHSIM Message File - PROLOG.msg  ---------------

S6F11  S6F11  W  E
        <L                              // 3
          <U4[4]=2>
          <U4[4]=999>
          <L                            // 1
            <L                          // 2
              <U2[2]=3>
              <L                        // 14
                <B[4]=x0a,x14,x1e,x28>
                <T[4]=T,F,F,T>
                <S1[4]=10,20,-30,-40>
                <U1[4]=50,60,70,80>
                <S2[8]=100,200,-300,-400>
                <U2[8]=500,600,700,800>
                <S4[16]=1000,2000,-3000,-4000>
                <U4[16]=5000,6000,7000,8000>
                <S8[32]=x0000000000002710,x0000000000004e20,xffffffffffff8ad0,xffffffffffff63c0>
                <U8[32]=x000000000000c350,x000000000000ea60,x0000000000011170,x0000000000013880>
                <E[16]=1.200000,2.300000,-3.400000,-5.500000>
                <D[32]=1.230000,2.340000,-3.450000,-5.560000>
                <A[17]="A1234567890ABCDEF">
                <J[17]="J1234567890ABCDEF">
              >
            >
          >
        >
        ¥

S6F12  S6F12
        <B[1]=x00>
        ¥
```