KaracrixBuilderV3 システムマニュアル

12章 制御プログラム

(章別取扱説明書 v1.10)

株式会社 エスアイ創房

KaracrixBuilder

■改定履歴

第1.00 版 2008/12/01

第1.10版 2015/08/10 プログラム編集エディタ選択の拡張(KBv3.50)

■ おことわり

- (1) 本書内容の一部又は全部を、無断で他に転載することは禁止されています。
- (2) 本書内容は、将来予告無く変更する場合があります。

KARACRIX は株式会社エスアイ創房の登録商標です。

KaracrixBuilderV3 システムマニュアル 第 1.10 版 © S.I.Soubou Inc.

KaracrixBuilder

目次

12 章	制御プログラム	12-1
12.1	制御プログラム登録画面の表示	12-1
12.2	プログラム作成手順	12-4
12.3	制御パラメータを使用するプログラム	12-9
12.4	プログラム編集	12-14
12.5	プログラムの DEBUG モニタ	12-21

S.I.Soubou Inc.

KaracrixBuilder

12章 制御プログラム

監視パネルやスケジュール機能を使用して、ポイントオブジェクトを手動で操作したり、ある程度自動的に操作したりすることができますが、さらに柔軟に各種オブジェクトの操作や取得データの情報処理などを行なうには、制御プログラムを作成することが必要です。

KaracrixBuilderシステムで作成された制御プログラムは、"プログラムオブジェクト"としてシステム上でポイントオブジェクトと同様の制御対象になります。プログラムから作成済みの"プログラムオブジェクト"を起動・停止したりすることも可能です。

制御プログラムでは、KCX ライブラリを使用することでオブジェクトの情報を取得したり、オブジェクトに操作コマンドを発行したりすることができます。KCX ライブラリは C 言語の関数ライブラリになっています。 KCX ライブラリの詳しい解説は「23 章 KCX ライブラリリファレンス」を参照してください。

本章ではプログラムを作成して実行するまでの手順を説明します。

12.1 制御プログラム登録画面の表示

メインメニューから"制御プログラム"を選択し、「制御プログラム登録」画面を開きます。 「制御プログラム登録」画面の機能を以下で説明します。

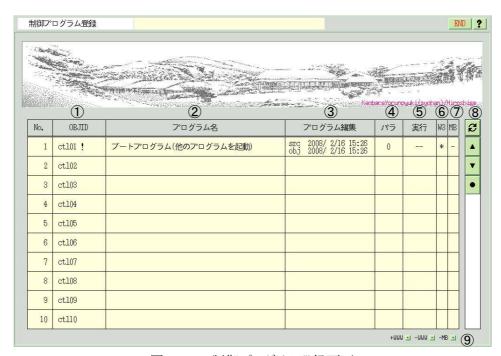


図 12.1.1 制御プログラム登録画面

S.I.Soubou Inc. 12-1

①OBJID

制御プログラムオブジェクトの管理 ID を表示しています。

1行目の"ctl001"には、"!"マークが付いています。これは、ctl001プログラムがシステム上特別なプログラムであることを示しています。"!"マークのプログラムは、KaracrixBuilder 起動時にシステムによって自動起動されるプログラム(ブートプログラムとも呼びます)です。但し、このプログラムは実行できるように予めコンパイルされている必要があります。KaracrixBuilder 起動と同時に ctl001 以外のプログラムを起動させたり、ポイントオブジェクトのデータをシステム起動時に任意に初期化したりする目的に使われます。

②プログラム名

プログラム名を登録します。(必須)

③プログラム編集

ソースプログラム作成日時、実行プログラム生成日時が表示されます。また、本欄をクリックすると"プログラム簡易編集"画面(12.4項)が表示されます。

④パラ

制御プログラムで使用している制御パラメータ数を表示します。

⑤ 実行

制御プログラムの実行状態を表示します(下表参照)。また、実行プログラムが生成されている場合には、項目をクリックするとプログラムの実行ダイアログが表示されます。

表 12.1.1 実行欄の表示

表示	意味
	プログラムが停止中
RUN	プログラムが実行中
INPUT	プログラムがユーザからのキー入力待ち(12章12.5で解説)

12-2 S.I.Soubou Inc.

6W3 (WWW)

7MB (MoBile)

登録されている制御プログラムをWebブラウザ(W3)、携帯端末(MB)からの参照及び設定を許可するかどうかの設定を行います。

本欄をマウスで選択してクリックすると設定ダイアログが表示されます。

設定には、以下の3つがあります。

新規登録した制御プログラムは、W3 及び MB のデフォルトは"-"になっています。



- : 非公開(Web または携帯端末から参照及び設定できません)

* : 一般ユーザ、管理者ユーザからの参照及び設定を許可します

S: 管理者ユーザからの参照及び設定を許可します

CAN: 設定をキャンセルします

図 12.1.2 アクセス許可設定ダイアログ

⑧編集

登録されている制御プログラムの消去、移動、コピー、エクスポートの編集を行うことが出来ます。

9+W3,-W3,-MB

登録されている制御プログラム画面の一括設定ボタンです。

"+W3"ボタンをチェックすると、⑥W3の設定を一括して全て許可(*)に設定します。

"-W3"ボタンをチェックすると、⑥W3の設定を一括して全て非公開(-)に設定します。

"-MB"ボタンをチェックすると、 ⑦MB の設定を一括して全て非公開(-)に設定します。

12.2 プログラム作成手順

(1)プログラム作成と取り込み

制御プログラムの作成手順としては、外部エディタ(gedit、vi、emacs 等)を使用して作成したプログラムを KaracrixBuilder のインポート機能を使用して取り込む方法と、KaracrixBuilder の簡易編集エディタで記述する方法がありますが、外部エディタを使用して作成したものをインポート(18章リソースファイルの入出力)する方法が効率的なので推奨します。その場合、作成したプログラム(拡張子".pgc")を"/tmp"ディレクトリに置いて下さい。

(2)プログラムの登録

メインメニューから"制御プログラム"を選択し、「制御プログラム登録」画面を開きます。何も登録されていないリスト上の"プログラム名"欄を選択してプログラム名称を最初に登録します。

※ "プログラム名"を登録しないと"プログラム編集"画面に進めません。

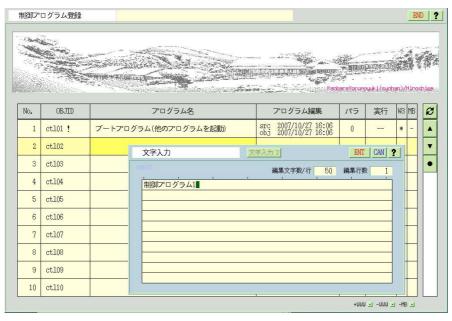


図 12.2.1 プログラム名称の決定

12-4 S.I.Soubou Inc.

(3)プログラムの作成編集

●外部プログラムをインポートする場合

外部エディタソフト等で作成した制御プログラムを"/tmp"ディレクトリに置いている場合には、プログラム名称を登録後に以下のようなインポートダイアログが表示されますので"YES"を選択します。"/tmp"ディレクトリ下に存在する"*.pgc"の拡張子のファイル一覧が表示されますので作成した制御プログラムファイルを選択して下さい。(18章 リソースファイルの入出力」参照)

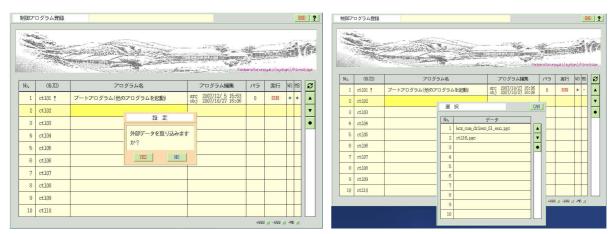


図 12.2.2 外部プログラムの取り込み編集

●インポートする外部プログラムが存在しない場合

インポートディレクトリ(/tmp)にプログラムファイルが存在しない場合は、「制御プログラム登録」画面の "プログラム編集"欄を選択して「プログラム簡易編集」画面を開き、プログラムを編集します。

なお、新規作成の場合には、以下の様に制御プログラムのテンプレートが編集画面に表示されます。

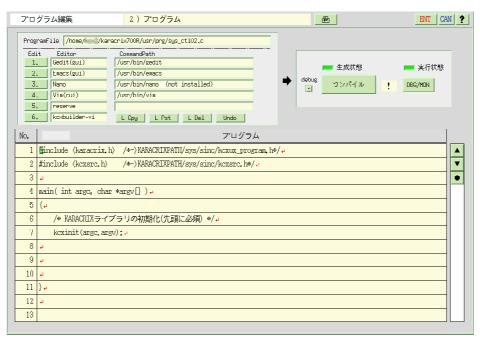


図 12.2.3 プログラム編集

S.I.Soubou Inc. ===

(4)プログラムのコンパイル

「プログラム簡易編集」画面の"コンパイル"ボタンを押すと、画面に表示されているプログラムソースコードのコンパイル作業が行なわれます。

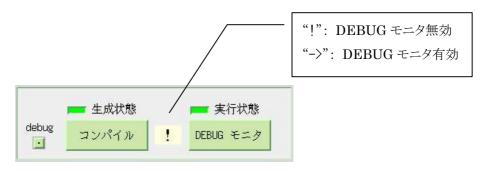


図 12.2.4 プログラムのコンパイル

プログラムの記述ミスなどで、コンパイルのエラーが発生した場合には、「コンパイル」画面にエラーメッセージが表示されますので、メッセージを参考にしてプログラムを修正し、エラーメッセージがなくなるまでコンパイル作業を繰り返します。

コンパイルが成功すると、"生成状態"インジケータが赤色に変化します。

※"!"マークは、「DEBUG モニタ」ボタンが無効であることを表します。debug ボタンについては、12.5 項で解説します(使用する場合には、解説をよく読んでからご使用下さい)



図 12.2.5 コンパイルエラー表示画面(サンプル)

(5)プログラム作成日の確認

プログラムの編集が終わり、コンパイルも正常に終了すると下図のように「制御プログラム登録」画面の "プログラム編集"欄に、作成したソースプログラム(src)と実行プログラム(obj)の生成された日付が表示されます。

プログラム編集	パラ	実行	МЗ	MB	2
src 2007/10/27 16:06 obj 2007/10/27 16:06	0	7 1111 1	*	-	
src 2007/10/27 16:08 obj 2007/10/27 16:08	0	4	=	=	

図 12.2.6 実行プログラム生成の日付

コンパイルを行なっていない場合、または、コンパイルが正常に終了しなかった場合には、実行プログラム(obj)の日付欄は下図のように空欄になります。

プログラム編集	パラ	実行	M3	MB	2
src 2007/10/27 16:06	0	(1114)	*	-	
src 2007/10/27 16:11 obj // :	0			-	

図 12.2.7 実行プログラムが作成されていない場合

(6)プログラムの実行

コンパイルが成功したら、「プログラム簡易編集」画面の"ENT"ボタンを押して「制御プログラム登録」画面に戻り、"実行"欄を選択して下さい。実行確認のため、「確認」ダイアログが表示されますので、"RUN"を選択するとプログラムを実行します。



図 12.2.8 プログラムの実行

なお、ここでコンパイルした実行ファイルは、以下の画面でも実行することができます。

- •「12.5 プログラムの DEBUG モニタ」で説明する「プログラム実行」画面
- ・制御パラメータ編集では、

「制御パラメータ」->「制御パラメータ入力」->「パラメータ編集」画面

・Web、携帯から操作を許可されたプログラムについては、

「パラメータ一覧」->「プログラム選択」->「パラメータ設定」画面

プログラムを実行すると「制御プログラム登録」画面の"実行"欄に、「RUN」と表示されます。

100			P	I.				
	2	ct102	制御プログラム1	src 2007/10/27 16:59 obj 2007/10/27 16:59	0 (RUN)-	22

12-8 **S.I.Soubou Inc.**

12.3 制御パラメータを使用するプログラム

プログラム中で使用する数値や文字列をパラメータとして、使用する環境や機能に合わせて変更して使いたい場合がよくあります。制御パラメータ登録機能を使用すれば、各種パラメータを定義して KCX ライブラリ関数から使用することができます。この機能により、プログラムを再コンパイルすることなくパラメータを変更するだけでプログラムの動作を変更することが可能です。

(1)制御パラメータ登録

「制御プログラム登録」画面の"パラ"欄を選択して下さい。「制御パラメータ登録」画面が表示されます。



図 12.3.1 制御パラメータ登録画面

①キーワード

制御プログラムが定義を検索するためのキーとして使用します。

パラメータのキーワードを英数文字で定義します。使用できる文字は、アスキー文字です。

②定義

キーワードに対応した値を設定します。設定された値は、すべて文字列として扱われます。整数値、実数値を設定してプログラム中で数値として扱いたい場合にはプログラム内での変換作業が必要です。 なお、本定義を設定できる画面は、本画面の他、以下の画面でも設定ができます。

制御パラメータ編集では、

「制御パラメータ」->「制御パラメータ入力」->「パラメータ編集」画面

・Web、携帯から操作を許可されたプログラムについては、

「パラメータ一覧」->「プログラム選択」->「パラメータ設定」画面

※画面例に関し後述「(4)パラメータ編集画面とWeb&携帯画面との競合設定について」参照

③入力画面:キーワード説明

制御パラメータ入力/パラメータ編集画面、Web、携帯画面に表示される項目の説明を入力します。

④入力画面: 備考

制御パラメータ入力/パラメータ編集画面、Web 画面に表示されるコメントを入力します。

(5)W3

Web ブラウザでパラメータ設定の表示を有効にするかどうかの設定です。 表示されたものは変更の対象となりますので、変更させたくないものは表示しないように設定します。

(6)MB

携帯端末からパラメータ設定の表示を有効にするかどうかの設定です。 表示されたものは変更の対象となりますので、変更させたくないものは表示しないように設定します。

⑦画面消去ボタン(CLEAR)

画面に表示されているパラメータを全て削除するときに使用します。

誤って削除したときは、画面の"CAN"ボタンで画面を終了します。"ENT"ボタンを押すと削除が確定してしまいますのでご注意下さい。

⑧インポートボタン(GET)

リソースファイル入出力ディレクトリ(/tmp)に存在する外部パラメータ定義ファイル(拡張子".pra")から、パラメータ定義を取り込みます。 画面がクリアされている状態でのみ有効です。

⑨エクスポートボタン(PUT)

画面上で定義したパラメータ設定を、リソースファイル入出力ディレクトリ(/tmp)にパラメータ定義ファイル (拡張子".pra")としてエクスポートします。 画面にデータが入力されている状態でのみ有効です。

12-10 **S.I.Soubou Inc.**

(2)制御パラメータ登録サンプル

制御パラメータ登録の例を以下に示します。



図 12.3.2 制御パラメータ登録画面例

S.I.Soubou Inc. 12-11

(3)制御パラメータを使用したプログラム

(2)で設定されている制御パラメータをプログラム中で使用する記述例を以下に示します。 このプログラムでは、パラメータのキーワードで MAX_value に設定されている値 9999 を取得して、計測値の上限を超えた場合の処理の条件分岐に使用しています。

(使用例)

```
/*プログラムパラメータ・キーワード数*/
int
         keywords;
                              /*プログラムパラメータ・キーワード格納*/
char
        *keyword [BUFSIZ];
        *paradata[BUFSIZ];
                              /*プログラムパラメータ・データ格納*/
char
                               /* タイムアウト値(秒) */
int
        timeout;
        mailperm;
                              /* メール送信許可フラグ */
int
        email1
                  [BUFSIZ];
                              /* 通知先メールアドレス 1 */
char
                              /* 通知先メールアドレス 2 */
char
        email2
                  [BUFSIZ];
char
        cpuname [BUFSIZ];
                              /* 装置名称 */
                              /* 上限值 */
float
        hi_limit_value;
                               /* 現在計測值 */
float
        current_value;
/* プログラムパラメータ取得関数 */
/* keyword, paradata のメモリ領域は関数内部で確保され、そのアドレスが戻されます */
keywords = kcxprg_para_data_get( 0/*mode*/, keyword, paradata, BUFSIZ );
/* キーワードの値を取得する */
for(i=0;i<keywords;i++){
 if( strcmp( keyword[i], "timeout" ) == 0 ){
     timeout = atoi( paradata[i] );
 if( strcmp( keyword[i], "cpuname" ) == 0 ){
     strcpy(cpuname, paradata[i]);
 if( strcmp( keyword[i], "mailperm" ) == 0 ){
     if(strcmp(paradata[i], "on") == 0){
         mailperm = 1;
     } else {
         mailperm = 0;
 if(strcmp(keyword[i], "1.email") == 0){
     strcpy( email1, paradata[i] );
 if(strcmp(keyword[i], "2.email") == 0){
     strcpy(email2, paradata[i]);
 if(strcmp(keyword[i], "MAX_value") == 0){ ← キーワード"MAX_value"の値を取得
     hi_limit_value = atof(paradata[i]);
 }
}
/* 取得した値を使用して処理を分岐する */
if ( hi_limit_value < current_value ) {</pre>
 /* 上限値を超えた場合の処理 */
```

12-12 **S.I.Soubou Inc.**

(4) パラメータ編集画面と Web &携帯画面との競合設定について

「制御パラメータ登録」画面で設定したパラメータは、以下の3つの画面でも設定することができます。 なお、制御パラメータ登録画面の場合と異なり、<u>定義(設定データ)のみの編集</u>という制限があります。また、 設定の競合にはご注意ください。

1. 「制御パラメータ入力」画面→「パラメータ編集」画面



図 12.3.3 「パラメータ編集」画面例

2. 「Web メニュー(パラメータ または 帳票)」→「パラメーター覧 または 帳票一覧」
→「プログラム選択」→「パラメータ設定」画面

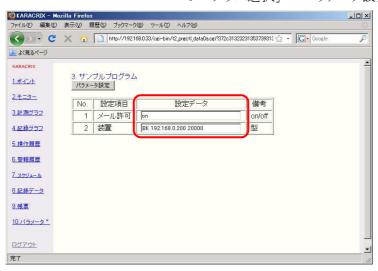


図 12.3.4 「Web パラメータ設定」画面例

3. 「携帯メニュー(パラメータ)」→「プログラム選択」→「パラメータ設定」画面

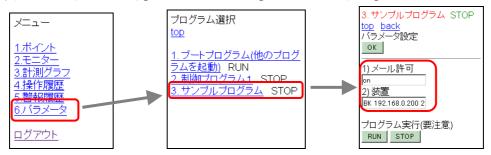


図 12.3.5 「携帯パラメータ設定」画面例

12.4 プログラム編集

プログラムの作成は、慣れたエディタ(gedit、vi、emacs など)を使用して行って下さい。エディタは、KaracrixBuilder 管理外のところで作成し、そのプログラムをリソースとしてインポート(取り込む)する機能 (「18章 リソースファイルの入出力」参照)を用いて KaracrixBuilder 内に取り込めます。また下記画面の編集ボタン(Edit:1.~6.)よりエディタを直接呼び起こすことも出来ます。

なお、プログラム実行中の場合、編集はできませんのでご注意下さい。

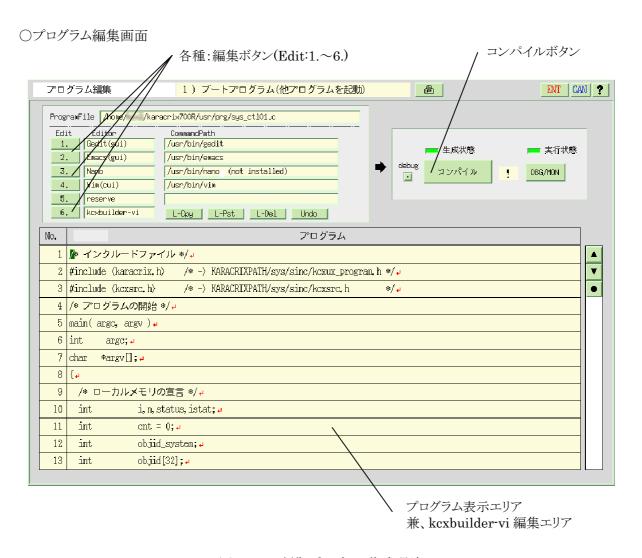


図 12.4.1 制御プログラム作成環境

12-14 **S.I.Soubou Inc.**

○プログラム編集ボタン

プログラム編集画面から呼び起こせるデフォルトエディタには、Gedit, Emacs, Nano, Vim があります。

なお、OSに該当エディタがインストールされていなかった場合、そのエディタには not installed が表示され使用できません。エディタの呼び起こし方は、例えば以下に示すデフォルト画面で、例えば Gedit を起動したい場合には、1.のボタンを押します。

エディタが、編集するファイルは、ProgramFile の欄に示されています。

実際に使用されるエディタコマンドは、CommanndPathの欄に示されています。

Editor 名の括弧内に示される gui, cui は、ユーザーインターフェイスを示しています。

6.の kcxbuilder-vi は、本編集画面を用いて vi ライクな簡易な編集する時に用います。

Program	File <mark>/home/ /ka</mark> n	acrix700R/usr/prg/sys_ctl01.c
Edit	Editor	CommandPath
1.	Gedit(gui)	/usr/bin/gedit
2.	Emacs(gui)	/usr/bin/emacs
3.	Nano	/usr/bin/nano (not installed)
4.	Vim(cui)	/usr/bin/vim
5.	reserve	
6.	kcxbuilder-vi	L-Cpy L-Pst L-Del Undo

図 12.4.2 プログラム編集ボタンと情報

◇編集ボタンのカスタマイズ

編集ボタン 1.~5.の割り当て及び起動端末を変更する場合は、\$KARACRIX/usr/env 内にある、syscnf.env ファイルの下記の部分を編集し、KaracrixBuilder を再起動します。

※上記\$KARACRIXとは、KaracrixBuilderがインストールされた位置を示します。例えば/home/karacrix700Rとかです。

```
ext_editor_1, GUI,"Gedit","/usr/bin/gedit",""
ext_editor_2, GUI,"Emacs","/usr/bin/emacs",""
ext_editor_3, CUI,"Nano","/usr/bin/nano.","gnome-terminal -e"
ext_editor_4, CUI,"Vim","/usr/bin/vim","gnome-terminal -e"
ext_editor_5, CUI,"reserve","",""
```

ext_editor_1~ext_editor_5は、ボタン1.~5.に対応し変更できません。

ext_editorの右辺に、GUIかCUIの表示、エディタ名、エディタコマンドパス、CUIの場合の起動端末を指定します。

ext_editor_5のreserveの部分は、お好きなエディタを定義するなど自由に使って構いません。

○エディタ起動例

以下にGUIエディタ(Gedit)及びCUIエディタ(Vim)を起動した時の画面例と注意点を示します。

□ GUI/Gedit 起動例

編集ボタン1.をクリックすると、KaracrixBuilderの管理画面が消えて、以下に示すGedit画面が現れます。 この画面でプログラムを編集します。編集が済みましたら、必ずファイルを『保存』してから、Geditを終了させて下さい。保存しなかった場合、編集されなかったものとして扱われます。Gedit の終了と同時にKaracrixBuilderの管理画面が戻って来ます。

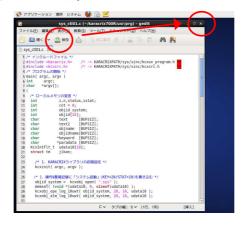


図 12.4.3 Gedit 起動状況

□ CUI/Vim 起動例

編集ボタン 4.をクリックすると、KaracrixBuilder の管理画面が消えて、以下に示す Vim 画面が現れます。 この画面でプログラムを編集します。編集が済みましたら、必ずファイルを保存してから、Vim を終了させて下さい。保存しなかった場合、編集されなかったものとして扱われます。

CUI 起動の場合、CUI 画面を終了させても自動的に KaracrixBuilder の管理画面が戻って来ません。CUI 画面操作終了後、「編集終了/確認」画面を"END"させて下さい。その後 KaracrixBuilder の管理画面が戻って来ます。

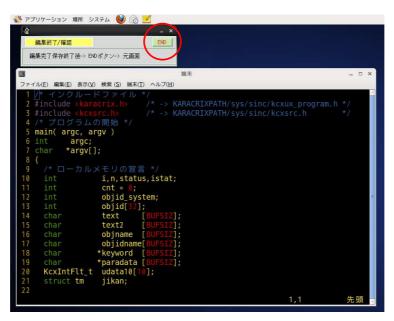


図 12.4.4 Vim 起動状況

○kcxbuilder-vi を用いた簡易な編集

6.のボタン凸を押すと"kcxbuilder-vi"表示が編集中を示す"..editing.."に表示が変わり、編集画面エリアを使用してテキストの入力や編集を行なうことができるようになります。

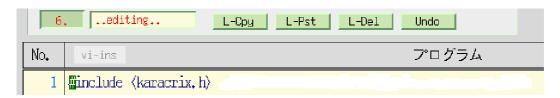


図 12.4.5 kcxbuilder-vi 編集モード

プログラム編集エリアでプログラムを作成します。

※以下の説明中で使用するカーソルの意味は、マウスカーソルではなく、編集領域での文字入力位置を示すカーソル(緑色反転)を示します

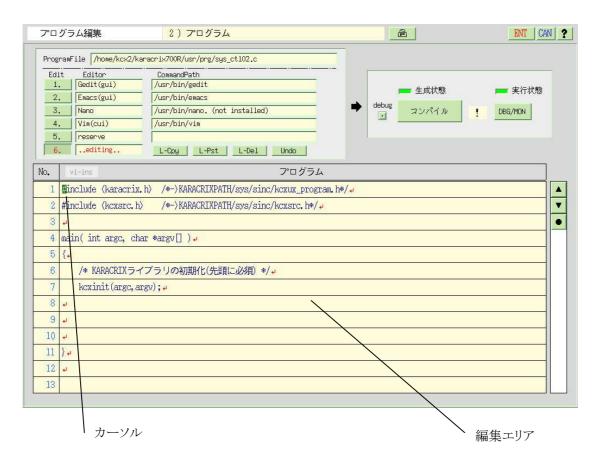


図 12.4.6 プログラム編集エリア

(1)文字の入力

文字はキーボードから入力します。最大編集行数は10000 行、1 行あたり半角で109 文字(改行含む)まで入力できます。

〇キ一操作について

文字以外のキー入力については以下のキー操作が有効です。

	X 11 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
機能	使用キー
改行	Enter
カーソルの移動	\rightarrow (右)、 \leftarrow (左)、 \uparrow (上)、 \downarrow (下)
スクロール	PageUp, PageDown
ジャンプ	Home(文頭へ)、End(文末へ)
文字の削除	Backspace (カーソルの前)、Delete (カーソル上)

表 12.4.1 キー操作の機能一覧

〇日本語の入力

日本語を入力したい位置にカーソルを移動して、そのカーソルの位置する同じ行をマウスでクリックすると下図のように「文字入力」ダイアログが表示されます。ここで日本語を入力します。

入力した日本語文字列は、ダイアログの ENT ボタンを押すことでカーソルの位置に挿入されます。

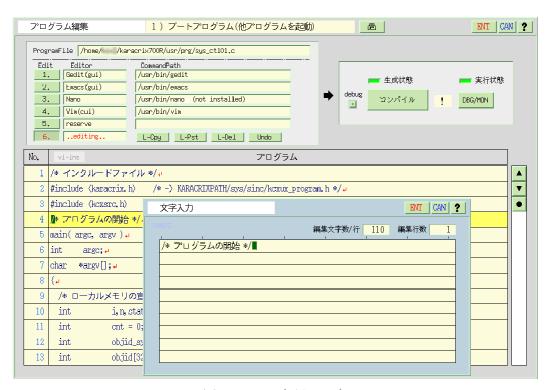


図 12.4.7 日本語の入力

[※]スクロールは、画面上のスクロールバーからでも行なえます。

(2)文字列の編集

○行をコピーする (L-Cpy) (L-Pst)

コピー元の行へカーソルを進め*、行コピーボタン(L-Cpy)を押します。

コピー先の行へカーソルを進め※、行貼り付けボタン(L-Pst)を押します。

※カラムの位置は問いません

○行を削除する (L-Del)

削除したい行へカーソルを進め*、行削除ボタンを押します。削除したものはバッファに残っていますのでコピーと同じように行貼り付けすることができます。

※カラムの位置は問いません

○操作をやり直す(Undo)

やり直しボタンを押すと編集操作前(1回前)の状態に戻ります。

(3)リスト印刷

■ボタンを押すとソースプログラムのリストを印刷します。

※プリンタの設定が正しくされている必要があります(「2章2.5 印刷環境の設定」参照)。

(4)画面テキスト編集

vi エディタライクな使い方ができます。使用方法は次のとおりです。 vi エディタについては LINUX、UNIX の解説書をご覧ください。

○編集モードと入力モードの切り替え

"編集"ボタンを押した"編集中"の初期状態ではテキストの色が濃紺色で表示され、入力モードになっています。編集モード表示エリアに以下のように入力モード(vi-ins)が表示されます。



キーボードからエスケープキー(ESC)を押すと<u>"入力モード"</u>から<u>"編集モード"</u>に切り替わります。編集モード表示エリアも以下のように入力モード(vi-cmd)が変ります。



編集モードにするとテキストは黒色表示になります。

○カーソルの移動

h,j,k,lキーが使用できます。

○やり直し

uで1回のみのやり直しができます。

○行コピー

yy で1行のみのバッファへのコピーができます。

○行削除

dd で1行のみの行削除ができます。削除した内容はバッファに残ります。

○行ペースト

p でバッファの内容をペースト(貼りつけ)します。

○一文字削除

xでカーソル上の一文字のみの削除ができます。削除した内容はバッファに残ります。

○改行

o でカーソルがある行の下に一行挿入ができます。

○編集モードへの切り替え(濃紺色表示)

iでカーソルの前に、aでカーソルの後から文字を挿入します。

(5)キーボード入力のトラブル

お使いのデスクトップ環境によっては、キーボードからの入力が受け付けられなくなる場合があります。これは、何らかのタイミングで、お使いのウィンドマネージャ等によって KaracrixBuilder のキー入力画面からキーのフォーカスが取られてしまう為です。この様な状態が発生しましたら、マウスポインター(カーソル)を画面から完全に外してから再度画面に入れて下さい。このタイミングでフォーカスが取り戻せてキー入力ができるようになるはずです。

12-20 **S.I.Soubou Inc.**

12.5 プログラムの DEBUG モニタ

「プログラム簡易編集」画面で、debug ボタンをチェックしてコンパイルを行うと、プログラム DEBUG モニタを使用して作成したプログラムの実行状況をソースレベルでモニタすることができます。ステップ実行も可能になっています。プログラムの制御ステップを1行ずつ確認できるのでデバッグ時に有効です。 DEBUG モニタボタンを押すと「プログラム実行」画面が表示されます。



"->": DEBUG モニタ有効

"!": DEBUG モニタ無効

デバッグコンパイルの使用上の注意

デバッグ機能を実現するために、プリプロセッサを使用していますが、KaracrixBuilder のプリプロセッサ処理能力は、現在のところ高くありません。従って、複雑な C 言語の構文を記述しますと構文解析不能で間違った結果を生成してしまう場合があります。コンパイルエラーが発生した場合に、ソースプログラムに文法上の間違いが見当たらないときには、構文を単純化してお試しください。

また、コンパイルが成功しても、アルゴリズムが正しくない場合もありますので十分にご注意下さい。 デバッグが終わって、運用で使用する場合には、必ず debug ボタンを解除してコンパイルした実行コード を作成して下さい。実行効率が上がるだけでなく、デバッグ機能追加のリスクが無くなり正常実行します。

(1)画面構成

下図のように操作部と実行モニター部およびデータ表示部の3つで構成されています。



図 12.5.1 プログラム実行モニタ画面

①操作部

6 つの操作ボタンで構成されます。ボタン上部にLED が配置され、プログラムの動作状態に応じて操作可(緑色)、操作不可(無色)、現在のプログラムの状態(赤色)で変化します。

②実行モニタ部

モニタ中のプログラムソースが表示されます。現在実行中の行が黄色で表示されます。 外部対話型入力コマンド(kexstr_scanfm()関数)を実行すると"入力待"欄に"*"が表示されます。

③データ表示部

KCX ライブラリの kcxstr_printm() 関数が出力したデータが表示されます。 ライブラリの使い方は「23 章 KCX ライブラリリファレンス」をご覧ください。

(2)プログラムの操作

プログラムの操作は操作部で行います。プログラムの実行の仕方は大きく分けてフリーラン(Free RUN)と、ステップラン(Step RUN)の2種類があります。フリーランはプログラムを連続して実行します。これに対しステップランは1行ずつ順々に実行します。

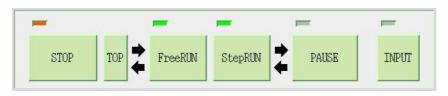


図 12.5.2 操作部

○フリーラン(FreeRUN)

FreeRIN ボタンを押すと、プログラムが停止している位置から順次実行します。

○ステップラン(StepRUN)

ボタンを押すと、プログラムが停止している位置から一行実行して一時停止します。ボタンを押すことで再び次の一行が実行されて一時停止します。この繰り返しで、一行毎に処理を確認しながら実行するときに使用します。

12-22 — S.I.Soubou Inc.

○プログラムの停止(STOP)

ボタンを押すと、プログラムを終了します。

○プログラムの一時停止 (PAUSE)

ボタンを押すと、プログラムを途中で一時停止します。

○データ入力 (INPUT)

プログラム実行中に、対話型入力コマンド(kcxstr_scanfm()関数)を実行すると、 ボタンの機能が有効になります。ここでボタンを押すと「文字入力」ダイアログが表示されますので、文字入力をすることにより実行中のプログラムに文字型データとして取り込むことができます。

○実行位置を先頭へ移動(TOP)

プログラム停止時に、 ボタンを押すと、プログラムの実行位置を先頭に戻します。

株式会社エスアイ創房 http://www.karacrix.jp