



CasePlayer2 チュートリアル

Ver. 1.0.4 2015/06/10

Cas	ePlayer2 チュートリアル	3
13	:じめに	3
	CasePlayer2 概要	3
	稼働環境	3
1.	CasePlayer2 起動	4
2.	新規プロジェクト作成	4
	その他の設定について(プロジェクトの新規作成ダイアログ)	6
	プロジェクトの種類	6
	オプションパラメータファイルの指定	6
	言語設定	6
	C ソースに関する設定	6
	C++ソースに関する設定	6
	アヤンブラソースに関する設定	6
	階層化を有効にする	6
	プロジェクトの設定データファイル	6
3	ソースコードの登録	0
0.	その他の設定について (ソースファイルの登録ダイアログ)	/
	サブフォルダも会む	0 8
	フーザインクルードパスに追加する	0 8
	コージインフルークハベに追加する	ο Ω
٨	ン) 「ル構成に日初とてンオルントとて豆螺ヶる	0 و
4.	1177音工次(保牛て「1)	10
	117% 言因見	10
	フャードリンフ版化	12
5) ビノノノノーへのノロー) ヤート	15
5.	計神解がて一て	15
	許啊胜们 这上	10
	11 恢音上风	10
	11111111111111111111111111111111111	17
~	2 数 変 数 の に し 、 し 、 し 、 、 、 、 、 、 、 、 、 、 、 、 、	19
6.		.20
	MISRA-し ルールナエックを美行	20
	表示オノンヨン	22
	ルールの選択	22
_	ソースコート・メトリックス閲覧	23
1.	反 余 傥 能	25
8.	· 川刷機能	27
9.	_ 仕様書の Word 変換	28
1(). 仕禄書の HTML 変換	30
1	. 詳細解析モードの詳細設定について	32
	フリフロセッサの設定	32
	#define (テファイン名 [=数値])	32
	#undef (デファイン名)	32
	プレインクルードファイル	32
	システムインクルードパス/ ユーザーインクルードパス	33
	Cオブションパラメータ	33
	User Keyword ユーザーキーワード	33
	置き換えキーワード	34
	Cオブションパラメータのその他の設定について	34
12	 最後に	34

CasePlayer2 チュートリアル

はじめに

この度は、仕様書作成・解析ツール、CasePlayer2 にご興味を頂き誠にありがとうございます。 本チュートリアルは、初めて CasePlayer2 をご使用頂くお客様にサンプルプログラムを使用しながら、 実際にツールをご操作頂き、基本的な使用方法と機能を理解して頂く事を目的として作成させて頂い ております。

チュートリアル開始の前に CasePlayer2 をインストールして頂きますようお願い致します。 CasePlayer2 はデフォルトでは"C:¥Program Files¥gaio¥Caseplayer2"にインストールされます。

CasePlayer2 概要

CasePlayer2 は、ANSI 準拠 C 言語、C++、組み込み向け C 言語(非 ANSI)を解析して、フローチャートなどの仕様書を作成するツールです。「仕様書ブラウザ」を搭載し、ソースコードと各仕様書間の連携を自由に取ることができます。過去のソースコードのロジック解析や、新たに開発したプログラムソースの仕様書作成作業を強力に支援致します。ソースコードの静的解析機能として、「外部変数」の参照/代入の一覧作成機能や、C 言語コーディング規約「MISRA-C」のルールチェック機能も搭載しています。



稼働環境

- * IBM PC/AT 互換機
- * USB ポート (ライセンスキーの装着に使用します)
- * Windows XP, Windows Vista (32 ビット), Windows 7 (32/64 ビット), Windows 8.1 (32 ビット、64 ビット)
- *インストールに必要な HDD 容量::80MB
- * HTML フォーマット文書の閲覧: Microsoft Internet Explorer 6 以上

* MS-Word 形式文書出力: Microsoft Word 2000 SP3 以降

- Microsoft Word 2000 SP3 以降 Microsoft Word 2002 SP3 以降 Microsoft Word 2003 SP3 以降 Microsoft Word 2007 SP2 以降
- Microsoft Word 2010 SP1 以降

1. CasePlayer2 起動

Windows のスタートメニュー 「全てのプログラム」 → 「CasePlayer2」 → 「CaseViewer.exe」をクリッ クして下さい。

🛹 XPS ビューアー	コントロール パネル
📑 デスクトップ ガジェット	デバノフ トプロンカー
💮 既定のプログラム	7/1/2/9/9-
🐌 activePDF	既定のプログラム
퉬 АТОК	
🕛 CasePlayer2	ヘルプとサポート
🔢 CaseViewer.exe	
🚰 チュートリアル	
🖟 FuncVarGrid 🗸 🗸	
◀ 前に戻る	
プログラムとファイルの検索 👂	シャットダウン ▶

2. 新規プロジェクト作成

ソースコード解析、仕様書生成実行のために CasePloayer2 の新規プロジェクトを作成しましょう。

- 1. CasePlayer2 メニュー「ファイル」→「新規作成」→「プロジェクト」をクリックして下さい。
- 2. プロジェクト名 : test
- 3. 場所: (任意)
- 4. プロジェクトの種類::仕様書生成のみ
- 5. 言語設定::ANSI-C
- 6. オプションパラメータファイルの指定:[アセンブラオプションパラメータファイル]にチェックを付けてく ださい。

[参照]ボタンで下記ファイルを指定してください。

- 7. アセンブラオプションパラメータファイル: C:¥Program Files¥gaio¥CasePlayer2¥sample¥Z80¥Z80.rev
- 8. 上記以外の設定はデフォルト設定のままにしてください。
- 9. [OK]ボタンをクリックしてください。

プロジェクトの新規作成	×		
プロジェクト名(P): test	ОК		
場所(L): C:¥Users¥Administrator¥D	Documents 参照… キャンセル		
補足情報(S):	ヘルプ(H)		
プロジェクトの種類	言語設定		
◎ 仕様書生成のみ(R)	Cソースに関する設定		
◎ 概要説明をソース(こ戻す(W)			
✓ ソースファイルをコピーする(Y)	ソースファイル拡張子: 。		
オブションパラメータファイルの指定	ヘッダファイル拡張子: h		
□ Cオプションパラメータファイル(C)	- C++ソースに関する設定		
デフォルト 参照…	🔲 C++ 解析を行う		
	ソースファイル拡張子: cpp.cxx.cc.c++		
▼ アセンブラオブションパラメータファイル(A)	ヘッダファイル拡張子: hpp hxx hh h++		
SasePlayer2¥sample¥Z80¥Z80rev 参照	アセンブラソースに関する設定		
	◎ 通常のアセンブラソース(ガイオ構造化マクロ)		
□ ソースファイルは絶対パスで管理する(F)	○ 構造化マクロ RA78k(日本電気 7k/0~4)		
 ご 階層化を有効にする 	○ 構造化マクロ SRA74(ルネサス 740/38000)		
	○ 構造化マクロ ルネサス H8/300,300H,H8S		
	◎ 構造化マクロ 富士通 FMC-8L		
	○構造化マクロ ルネサス M16C		
	ファイル拡張子(T): xas		
□ プロジェクトの設定データファイル:	参照]		
ターゲットCPU及びコンパイラ固有の設定	,		

上記、3.場所: (任意)で選択して頂いたパスにプロジェクトフォルダが作成され、そのフォルダ配下に はプロジェクトファイル(test.vproj)と関連ファイルが配置されます。test.vprojをダブルクリックする事 で CasePlayer2 を起動する事が可能です。



その他の設定について(プロジェクトの新規作成ダイアログ)

プロジェクトの種類

仕様書生成のみ::本モードではソースコードから各種仕様書を生成する事ができます。 (チュートリアルではこちらを使用します)

概要説明をソースに戻す:本モードでは上記の[仕様書生成のみ]に加え、仕様書に書き込んだコメントをソースファイルのコメントとしてソースに戻すことが出来ます。[ソースファイルをコピーする]が ON になっている場合、コメントの反映はプロジェクト内にコピーされたソースに反映されます。

オプションパラメータファイルの指定

Cオプションパラメータファイル: Cオプションパラメータファイルは C 言語のコンパイラ依存の表記などを定義します。 ANSI-C 規格に準拠している場合には設定する必要はありません。

アセンブラオプションパラメータファイル

アセンブラオプションパラメータファイルは主にアセンブル言語の命令を定義します。 ガイオでは 40 以上のマイクロプロセッサのパラメータファイルをサンプルとしてご提供しております。 本チュートリアルでは Z80 マイコンのアセンブラ解析に使用致します。

言語設定

Cソースに関する設定

C の言語仕様を設定してください。本チュートリアルでは[ANSI-C]を使用します。

C++ソースに関する設定

C++解析を行う場合は、「C++解析を行う」のチェック指定を有効にしてください。本チュートリアルでは 使用致しません。

アセンブラソースに関する設定

アセンブリマクロ記述を解析するための種類を選択して下さい。本チュートリアルでは[通常のアセン ブラソース(ガイオ構造化マクロ)]を使用します。

階層化を有効にする

チェックを ON にした場合、プロジェクト内にフォルダを作成し、その中にソースファイルを登録することができます。本チュートリアルでは使用致しません。

プロジェクトの設定データファイル

既存のプロジェクトの設定を新規に作成するプロジェクトに設定できます。設定を使用する場合は、あらかじめプロジェクトから設定データをエクスポートしておく必要があります。 本チュートリアルでは使用致しません。

3. ソースコードの登録

プロジェクト作成後、次に解析対象のソースコードをプロジェクト登録します。

1. プロジェクトビューのプロジェクト名(test)を右クリックして、

「ソースファイルの登録」→「フォルダから」をクリックして下さい。

Caseviewer	V5.6.2						
ファイル(F) 表示(V) プロジェクト(P) ツール(T) ヘルプ(H)							
🗃 🚅 🔐 🕹 📾 🔜 📾 🔜 🕅 🗑 📓 🗐 😭 🐼 🖬 🖵 🔲 🗃 🖉 🤞							
	ソースファイルの登録(A)	•	ファイルから(F)				
⊡- <mark>=≥ test</mark>	ソースをヘッダファイルとして登録(G)	•	フォルダから(D)				
	プロジェクト内文字列検索						
	仕様書生成(C)						

2. チュートリアル用のソースが置かれている下記フォルダを選択して下さい。 C:¥Program Files¥gaio¥CasePlayer2¥sample¥チュートリアル

拡張子(.xas)のファイルはアセンブラソースになります。本チュートリアルでは z80 用のアセンブルソ ースを使用します。アセンブラソースとして認識する拡張子の設定は CasePlayer2 メニュー 「プロジ ェクト」→「プロパティ」で変更する事が可能です。

ソースファイルの)登録 (フォ	トルダ指定)	1		×
フォルダ(D): C	≿¥Program	Files¥gaio¥CaseP	layer2¥sample	¥チュートリアル	参照
サブフォルダ コマルク・	(も含む(S)	<u>ר</u> ב –	ザーインクルード。	パスに追加する	6(P)
ノ4ルター ▼Cソースファ4	(JV(C)	▼ アセンブラソース	.ファイル(A)	🔽 ヘッダフ	ァイル(I)
0++9-27	アイル	C++ \v\$771			
.¥etosc		↓ 更新(V)		
 ✓ .¥kconv.c ✓ .¥sample.xa ✓ .¥stoe.c 	IS				選択(0)
					きて選択(L)
					₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
•				•	
□ファイル構成(こ合わせてフォ	ゖルダ付きで登録す	5		
	ОК) <i>キャン</i>	211	へルプ(H)	

3.「OK」ボタンをクリックして下さい。

その他の設定について (ソースファイルの登録ダイアログ)

サブフォルダも含む

このチェックボタンを ON にした場合、サブフォルダ内のソースファイルもファイルリストに表示する事ができます。

ユーザインクルードパスに追加する

このチェックボタンを ON にしタ場合、ヘッダファイルを登録する際、そのファイルのあるフォルダパスを プリプロセッサ設定のユーザーインクルードパスに追加する事ができます。

ファイル構成に合わせてフォルダ付きで登録する

このチェックボタンを ON にしてサブフォルダ内のファイルを登録した場合、プロジェクトにサブフォルダ と同名のフォルダが自動的に作成され、そのフォルダ内にファイルを登録する事ができます。 ただし、このチェックボタンは、プロジェクトの新規作成時に「階層化を有効にする」をチェックしている 場合のみ有効です。

4. 仕様書生成(標準モード)

それでは標準モードで仕様書を生成してみましょう。本モードではフローチャート、モジュール仕様書などを生成する事ができます。

1. アプリケーションメニュー「プロジェクト」->「全仕様書生成」.をクリックしてください。 ソースファイルを解析し各種仕様書の生成を開始します。



2. 生成された各種仕様書はプロジェクトビュー内のツリーの[仕様書]を展開して頂くと表示する事が できます。



標準モードでは下記の仕様書が生成可能です。 フローチャート: 関数の構造をフローで表示する仕様書です。 モジュール仕様書: 関数の引数、戻り値などを表示する仕様書です。 モジュール構造図: 関数の呼出し関係を表示する仕様書です。 構造体仕様書: 構造体のメンバなどを表示する仕様書です。 モジュール一覧:解析した関数の一覧を表示する仕様書です。 構造体一覧: 解析した構造体の一覧を表示する仕様書です。

上記以外のドキュメント生成は詳細解析モードで生成する必要があります。(本チュートリアル後 半で説明します).

仕様書閲覧

モジュール構造図を見てみましょう

- 1. プロジェクトビューの[仕様書]→[モジュール構造図]を選択して下さい。
- 2. 仕様書リストビュー内にモジュール仕様書の一覧が表示されますので、

Test.vblk をダブルクリックして下さい。

🖃 😫 test	名前	ソースファイル名
	📲 test.vblk	<プロジェクト全体>
⊨… □ ソースファイル	📲 etos.vblk	etos.c
C etos.c	Rconv.vblk	kconv.c
kconv.c	📲 sample.vblk	sample.xas
C stoe.c	📲 stoe.vblk	stoe.c
A sample.xas		
チャート		
🖬 モジュール仕様書		
11 モジュール構造図		

表示倍率、表示モードの変更などが可能です。

№ モジュール構造図ビューア - [test.vblk]	A Laboratory		
10 ファイル(F) 編集(E) 表示(V) ツール	レ(T) ウィンドウ(W) ヘルプ(H)		_ & ×
🎽 🎽 🖓 🎧 🔛 🛅 🗈 🖻 🗎	🛧 🎒 🖻 🧇 💡	表示モード コメント ▼ 表示倍率 60%	QQ
main 標準入力から標準出力へ漢字コー 支援を行う	exit 異常終了 Stricep -政したか? stoe SJISからEUCに変換	igstohar i文字標準入力から i文字を標準出力へ i文字を標準出力へ	

チャートリンク機能

簡単にソースコード⇔仕様書ドキュメント間をジャンプできます。 フローチャートの右クリックメニューからソースコードの該当箇所へジャンプしてみましょう。

1. プロジェクトビューの[仕様書]→[フローチャート]を選択して下さい。
 2. 仕様書リストビュー内にフローチャートの一覧が表示されますので、

etos をダブルクリックして下さい

ファイル(F) 編集(E) 表示(V) ツール(T) ウィ	rンドウ(W) ヘルプ(H)		_ 8
a 🖪 🖷 🖀 🖌 🖻 🖂 🛱 🖊	🖻 🧇 🔋	表示モード 処理概要+	プログラム
表示倍率 40%	-		• Q
1次15日2日本人力が65 if ((c1 - setthar()) = −1) FALSE			
TRUE			
()			
外事つうびセット patii - 1:			
115(25) 1 7 202 15 [21 2 - 0.01] FALSE			
if ((.2 - settar())1)		### 55 55 50 1 if (cl = 0,5e)	- Thicks
		IRUE	
		if ((c2 - set char 0) 0.ff)	FALSE
FALSE		RVE	-
II (CDA)		(exit(2):	2
JR05 cZ = cZ=0xa1+0x40:	c2 = c2-0.m1+0.9f:	outchar .	1
		DUP 200 (2);	Ш
if (c2 > Qx7e)			
			•

- 3. 関数 etos のフローチャートが表示されます。表示倍率は、プルダウンメニューから変更する事が可 能です。
- 4. チャートボックスを選択して、右クリックメニュー→「ソースを開く」をクリックして下さい。



```
選択したチャートボックスに対応したソースコードの行が表示されます。.
```



仕様書とソースコードとが互いにリンクされているので、容易に右クリックメニューからプログラム文書 とソースコードを開くこと.ができます。

仕様書ビューア - [モジ	ユール仕様書 - etos]				
_ ファイル(F) 編集(E) □ □) 表示(V) ツール(T) ワイントワ(W) ヘルノ(H)		_ é		
▋□					
モシュール住	「依書				
名称 et os		種別	記憶クラス		
		Normal	extern		
定義位置 etos	切り取り(T)	Ctrl+X			
概要説明	コピー(C)	Ctrl+C			
EUCからSJISに変	貼り付け(P)	Ctrl+V			
	削除(R)	Del			
	引数の挿入(U)				
	引数の追加(D)	E			
戻り値	仕様書ブラウザを開く(B)	-			
<u>뽀</u>		-			
	ソースを開く(5) フローチャートを開く(5)				
ローカル変数一覧	フロ ブヤ 「ご開く(I) モジュール/メンバ期数仕様書を問く(M)				
変数名	モジュール構造図【大】を聞く(G)	-			
c1	モジュール構造図【小】を開く(N)	-			
c2	モジュール一覧/メンバ関数一覧を開く(L)	-			
gaiji	ファイル内モジュール一覧/メンバ関数一覧を開く(I)	E			
呼び出し関数一覧	構造体/クラス仕様書を開く(D)				
関数名	構造体/クラス一覧を開く(K)		行番号		
getchar	外部変数一覧を開く(V)	_	14		
exit	typedef一覧を開く(T)	-	18		
putchar	#define一覧を開く(O)	-	36		
補足説明	外部変数リファレンスリストを開く(E)				
	関数/メンバ関数リファレンスリストを開く(R)				
	構造体/クラスリファレンスリストを開く(A)				
	メンバ/メンバ変数リファレンスリストを開く(B)				
	C++仕様書(Z)	+			
コーチャートを開く	WinAMS連携	• •	書き		

Ε

1 モジュール構造図ビューア - [test.vblk]			
📲 ファイル(F) 編集(E) 表示(V) ツール(T) ウィンドウ(W) へ	ルプ(H)	-	
🖆 M A A A 🗌 🖹 🖻 🖻 🖻 🖉 🖉 🎗	表示モードコン	メント	▼ 表示倍率 60% ▼
Imain			
stoe	getchar		
SJISからEUCI実践	仕様書ブラウザを開く(B)		
	定義位置を表示(D)	•	ソースを開く(S)
	参照位置を表示(R)	•	フローチャートを開く(F)
	モジュール/メンバ関数仕様書を開く(M)	Ī	
	モジュール一覧/メンバ関数一覧を開く(I)		
	ファイル内モジュール一覧/メンバ関数一覧を開く(L)		
	関数/メンバ関数リファレンスリストを開く(U)		
図 2 モジュール構	造図からソースコードへ移	動	

アセンブラソースのフローチャート

CasePlayer2 はアセンブラソースからも仕様書を生成する事が可能です。本チュートリアルでは Z80 デバイスのアセンブラソースを含んでいます。

1	;		
2 3 4	;	ブイ I - ・	ログ Z 8 0 4 相ステッピングモータの加減速運転プログラム
5 6 7 8	N S PTO	EQU EQU EQU	200 ;Number of steps before reaching slew 399 ;2*N-1 0COH ;ポート O
10	PROG	i	SECT CODE, ORG=9000H
12 13 14 15 16	STAR	T: LD LD OUT EXX LD	LD B,33H ;データ出力(0011) C,0FDH ; A,B (PTO),A ; i励磁データの待避 IY,Y ;コマント*テーブルのアドレス設定
19 20 21 22 23	LOOP	1: LD LD OR JP	LD E,(IY+O) ;ステップ数の入力 D,(IY+1) ; A,D ; E ; Z,STOP ;コマンド継続中か?
25		LD Add	A,D A,O
27 28 29 30 31	; PLUS	: NEG	JP M,MINUS ; 回転方向はCWか? NEG ; CW処理 ID A.O :
32 33 34 35	MINU	s:	JR.X CLEAŔ CPL ;CCW処理 ID D.A :

Sample.xas のフローチャート表示方法:

- 1. プロジェクトビュー(ファイルタブ)の[ソースファイル]→[Sample.xas]を選択して下さい。
- 2. 仕様書リストビュー内に表示される START (フローチャート)をダブルクリックして下さい。



3. アセンブリルーチン START のフローチャートが表示されます。

(注釈)::アセンブラ命令は MPU に応じて異なるため(P5 参照)、プロジェクト生成時に 適切なアセンブリパラメータファイル(*.rev)を設定する必要があります。



設定して頂く(*.rev)ファイルは C:¥Program Files¥gaio¥CasePlayer2¥sample 配下にいくつかのサンプルがございます。

. 作成された仕様書が解析した MPU のフローとして正しく表示されない場合、(*.rev) ファイルをユーザー様で変更して頂く必要がございます。

5. 詳細解析モード

詳細解析モードでは、外部変数一覧、構造体リファレンスリストなどの仕様書を追加で作成する事が可能です。これらの仕様書は、外部変数の予期せぬ書き込みや参照、構造体の構造上の問題な どを早期に発見するのに役立ちます。

詳細解析モードでは、全てのコンパイルソースとヘッダファイルを解析のためにおプロジェクトに追加 する必要がございます。本チュートリアルはシンプルな構成ですので、詳細解析のために追加するフ ァイル、設定はございません。実際の組み込みプロジェクトでは詳細解析のために、CasePlayer2の 設定項目「プリプロセッサの設定」、「Cオプションパラメータの設定」をして頂く必要がございます。 これらの設定につきましては、チュートリアルの項目11以降にてご紹介させて頂いておりますのでご 参照ください。

詳細解析モードでは標準モードの仕様書に加えて、下記の仕様書が生成可能です。

外部変数一覧 typedef 一覧 #define 一覧 インクルード関係図 外部変数リファレンスリスト 関数リファレンスリスト 構造体リファレンスリスト メンバリファレンスリスト

詳細解析設定

詳細解析モードの設定:

- 1. [プロジェクトビュー]の[設定]タブを表示して下さい。
- 2. [仕様書生成]をダブルクリックして下さい。



~ ~

- 3. [詳細解析]の配下[winAMS 用プログラム解析情報を生成する]以外のすべてを ON にしてください。
- 4. [OK]ボタンをクリックして下さい。

仕様書生成の設定	LOW SECTION AND	
 仕様書生成の微調整 処理ボックスへの編集 コメントが出現する毎(こ(C) スペース行が出現する毎(こ(S) 1行毎(こ(O) 	処理概要の抽出 □ コメント行から抽出する(L) ○ こちらを優先 ○ 後続コメントから抽出する(A) ◎ こちらを優先 アセンブラ仕様書生成の自動条件 ○ コメント生成よりも後続コメントを優 先的に抽出する(Y) □ プログラムのコメントアウトを処理概要にしない Dコメントから抽出する	詳細解析 √ 詳細解析を行う(D) オプション生成 √ #define一覧(ブラウザ情報)を生成する √ 外部変数リファレンスリストを生成する √ 関数リファレンスリストを生成する √ 構造(本リファレンスリストを生成する √ メンバリファレンスリストを生成する ~ メンバリファレンスリストを生成する
 漢字コード ジロ ③ SJJS(J) ③ 入力EUC,出力SJJS(E) ⑦ 仕様書のコメントの行頭の空白をかい ⑦ 構造体のタヴ名にtypedef名を使用: 図 エラー出力にワーニングを出さない(W) 重大なエラーでない限り生成を続行: □ long longを有効にする □ wchar_tを予約語とする □ forループスコープの強制準拠 	ジェクト管理ファイル プロジェクト内でユニークな名前をつける(P) ソースファイル内でユニークな名前をつける(Q) 小する VC++解析設定 する ユニットマスター用プログラム解析情報を生 成する ア素型テストデータ生成を行う ランタイム型情報を有効にする ブリコンパイル済みヘッダを使用する 設定… 環境選択 ③ VisualStudio 2005/2008	 ✓ 外部変数を一覧に出力する モジュール仕様書の情報追加 ✓ 参照している外部変数一覧 ✓ 代入している外部変数一覧 ✓ ペンしている外部変数一覧 ✓ ペンしている外部変数一覧 ✓ 呼び出し関数一覧 ✓ 呼び出し関数一覧 ✓ winAMS用プログラム解析情報を生成する ○ 株米エリストナータエがどロフ ○ General ードを生成する ○ スケルトン生成のための情報を生成する ○ 埋め込みコード生成
	OK キャンセル	仕様書生成前にカバレッジ測 □ 定用オブジェクトビルド環境に 設定… ファイルをコピーする この設定を既定値にする

仕様書生成

CasePlayer2 メニュー「プロジェクト」→「全仕様書生成」.をクリックしてください。 ソースファイルを解析し各種仕様書を生成を開始します。

プロ	Iジェクト(P) ツール(T) ウィンドウ(W) ヘルプ(H)	
	プロパティ(R)	
	ソースファイルの登録(A)	F
	ソースをヘッダファイルとして登録(G)	×
	ソースファイルの削除(D)	
	プロジェクト内文字列検索	
	仕様書を生成(C)	
	全仕様書生成	
	リファレンスリストの生成(R)	

仕様書閲覧

外部変数一覧と外部変数リファレンスリストを見てみましょう。 作成したプログラムの仕様書を表示する操作方法は、前述の標準モードと同様になります。

- 1. プロジェクトビュー(ファイルタブ)を表示して下さい。
- 2. [仕様書]→[外部変数一覧]を選択して下さい。
- 3. 仕様書リストビュー内に表示される test.vvlt をダブルクリックして下さい。

2	3 仕様書ビューア - [外部変数一覧 - test.vvlt]										
	☑ ファイル(F) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)										
	🗅 🖆 🖬 🖞 X 🕫 🐻 🖹 🚇 🐘 🚑 🚇 🧇 🎖 🛇										
	表示モード 全表示 → 表示順 名称順 →										
	名称	記憶クラ ス	型	初期化記 述	定義ファイル名	行番 号	コメント				
	aaa	extern	int		etos.c	51					
	kcode extern struct cod e_table [1] kconv.c 13 漢字コードの名前 指定を柔軟にする										
V											

- 4. Kcode を選択して右クリックして下さい。
- 5. 「外部変数リファレンスリストを開く」をクリックして下さい。



外部変数リファレンスリストは kcode が参照されている位置をソースファイル名、行番号、関数名 を含むリストで表示します。リストの上で右クリックするとメニューが表示され、Kcode が参照 されている各種仕様書にジャンプできます。

2	仕様書ビューア -	[外部変数リファレンスリスト - kcode]		x	仕様書ブラウザを開く(B)			
	ファイル(F) 表] 🗃 🖬 🐰 🖣	示(V) ツール(T) ウィンドウ(W) 3 💼 🗙 い 🗃 🖺 🗟 胀 é	ヘルプ(H) ቜ 🚇 🤣 🤋	-	Ξ×	ソースを開く(S) 宣言位置を開く(Q)			
	外部変数!	リファレンスリスト			Â	フローチャートを開く(F) モジュール/メンバ関数仕様書を開く(M)			
	名称	kcode				モジュール構造図【大】を開く(G)			
	記憶クラス	extern				モンユール構造図【小】を開く(N) エミュール一般(ハン・ビリ数一般を思く(1)			
	型	struct code_table [14]				モンエールー見/メノハ関鉄一見を開く(L)			
	定義位置	kconv.c 13行目				ノアイル内モンユール一見/ メノハ関数一見を開く(I)			
	コメント	漢字コードの名前指定を柔軟にす	する			備這4/ クラス(11 快音を開く(D) −			
	参照位置一覧				≡ .				
	参照ファイル名	1	行番号	関数名	Į.	外部変数一覧を開く(V)			
	kconv.c		41	main		typedet一見を開く(I)			
	kconv.c		42	main	÷	#denne一見を開く(U)			
	kconv.c		43	main		21 印装数リファレンスリストで開く(E) 明教 ノット, ご明教 ロコート・フロフトを明く(D)			
	kconv.c		47	main		関鉄/ スノハ関鉄リファレンスリストを開く(N)			
	kconv.c		48	main		備連体/ クリスリファレンスリストで用ヽ(A)			
	kconv.c		49	main		メンハン メンハ変動リンドレンベリストで用く(D)			
					-	C++仕様書(Z)			
レラ	f1			*	禁				

変数参照情報の閲覧

詳細解析モードでの仕様書生成によりモジュール仕様書に追加された変数の参照情報を見てみましょう。

- 1. プロジェクトビュー(ファイルタブ)の[仕様書]→[モジュール仕様書]を選択して下さい。
- 2. 仕様書リストビュー内に表示される main をダブルクリックして下さい。

詳細解析モードで仕様書を生成した場合、次の項目が追加されます。 参照している外部変数一覧,ローカル変数一覧,呼出し関数一覧.

	レ仕様書							_
名称 main 種別 記憶クラス Normal extern								
定義位置 kc	onv.c 32行	E						
概要説明								
標準人力から得	『準出力へ演	7-1	変換を行う					
戻り値								
型	コメン	+						
void								
引数								7
名称		Į	型			コメント		1
argc		i	int			コマンドパラメー	-夕の数	1
argv		c	char ∗∗			コマンドバラメ [、] ボインタ	- タテーブルの	1
参照しているが								
SUC C 1 2 2 1	h部安奴一覧							
変数名	*都変致一覧		定義ファ	イル名			行番号	
変数名 kcode	*部変数一覧		定義ファ kconv.c	イル名			行番号 13	
変数名 kcode ローカル変数-	►部変数一覧 覧		定義ファ kconv.c	イル名			行番号 13	
変数名 kcode ローカル変数- 変数名	►部妥奴一覧 -覧	型	定義ファ kconv.c	イル名		メント	行番号 13	
変数名 kcode ローカル変数- 変数名 i	►部妥奴一覧 ─覧	型 int	定義ファ kconv.c	イル名 記憶クラス auto		メント	行番号 13	
変数名 kcode ローカル変数一 変数名 i from_code	►部変数一覧 - -覧	型 int int	定義ファ kconv.c	イル名 記憶クラス auto auto		メント	行番号 13	
変数名 kcode ローカル変数一 変数名 i from_code to_code	↑節変数一覧 - - 覧	型 int int int	定義ファ kconv.c	イル名 記憶クラス auto auto auto	3	メント	行番号 13	
a mage Consol 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一	↑節変数一覧 - - 覧 -	型 int int int	定義ファ kconv.c	イル名 記憶クラス auto auto auto		メント	行番号 13	
auto Consol 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名	↑節変数一覧 覧 覧	型 int int int	定義ファ kconv.c	イル名 記憶クラス auto auto auto		メント	行番号 13 行番号	
auto Consol 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名 etos	·節炎奴一與 -覧 -覧	型 int int int	定義ファ kconv.c	イル名 記憶クラス auto auto auto		メント	行番号 13 行番号 10	
auto Constraint 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名 etos exit	·部变数一覧 -覧 -覧	型 int int int	定義ファ kconv.c 定義ファ etos.c etos.c	イル名 記憶クラス auto auto auto - イル名		メント	行番号 13 行番号 10 18	
auto Constraint 変数名 に ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名 etos etos etos etos etos etos	*節炎奴一覧 -覧 -覧	型 int int int	定義ファ kconv.c 定義ファ etos.c etos.c kconv.c	イル名 記憶クラス auto auto auto イル名		メント	行番号 13	
a mice Consol 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名 etos etos exit stricmp copy	*部変数一覧 -覧 -覧	型 int int int	定義ファ kconv.c 定義ファ etos.c etos.c kconv.c	イル名 記憶クラス auto auto auto		メント	行番号 13 行番号 10 18 42 71	
auto Constraint 変数名 kcode ローカル変数一 変数名 i from_code to_code 呼び出し関数一 関数名 etos exit stricmp copy stoe	·節炎奴一與 - 覧 - 覧	型 int int int	定義ファ kconv.c 定義ファ etos.c etos.c kconv.c kconv.c	イル名 記憶クラス auto auto auto		メント	行番号 13 行番号 10 18 42 71 3	

これにて、本チュートリアルでの仕様書生成についての説明は終了いたします。 ここまでの手順にて、標準モードと詳細解析モード、各種仕様書間の右クリックメニューでのリンク機能 についてご説明させて頂きました。

6. MISRA-C ルールチェック

MISRA-Cは、組込みソフトウェアのコードの安全性、移植性と信頼性を容易にするために、MISRA (自動車産業ソフトウェア信頼性協会)によって開発された C プログラミング言語のためのソフトウェア 開発標準です。

ここでは、CasePlayer2 の MISRA-C ルールチェック機能についてご説明させて頂きます。 CasePlayer2 にてサポートしている MISRA-C ルールセットは MISRA-C1998 及び MISRA-C2004 になります。

MISRA-C ルールチェックを実行

MISRA-C ルールチェックを行うためには、本チュートリアルの前篇でご説明させて頂きました 詳細解析モードの設定が必要になります。これ以降のご説明は、前篇で使用致しました test.vprojの 詳細解析モードの設定が行われている事を前提にご説明させて頂きます。

1. アプリケーションメニュー「プロジェクト」->「全 MISRA-C 解析」.をクリックしてください。

MISRA-C解析(E)
全MISRA-C解析
設定データのエクスポート
設定データのインポート

2. プロジェクトビュー(ファイルタブ)の[MISRA-C 解析情報]を開いて下さい。



3. [MISRA-C:2004 解析結果]をダブルクリックして下さい。

☑ 仕様書ビューア - [- test.mmd4]	レンドウ		X					
MISRA-C:2004 解析結果		表示メッセージ数:161/総メッセージ数:161						
ルール番号[1.2] 未定義又は未	親定の	動作に依存してはならない。 メッセージ数:1						
ファイル名	行番号	補足説明						
kconv.c	77	ファイルが改行文字で終了していない						
ルール番号[3-3] 選定したコン 文書化し,+	パイラ(*分に配!	り整数除算の実装について確認し, メッセージ数:2 奮すべきである。]					
ファイル名	行番号	補足説明						
etos.c	24							
	31]					
ルール番号[5.7] 識別子は,再	使用す	べきではない。 メッセージ数:5]					
ファイル名	行番号	補足説明						
etos.c	11	動的変数が他の関数で使用されている(c1)						
		動的変数が他の関数で使用されている(c2)						
kconv.c	72	動的変数が他の関数で使用されている(c1)						
stoe.c	4	動的変数が他の関数で使用されている(c1)						
		動的変数が他の関数で使用されている(c2)	J					
ルール番号[6.3] 基本型の代れ 示すtypedef)りに,† を用いな	サイズ及び符号属性(signedness)を メッセージ数:14 ければならない。]					
ファイル名	行番号	補足説明						
etos.c	4		•					
レディ			書禁					

MISRA-C のルールに違反しているコードがある場合、違反しているルール番号と共にソース名、行番号、補足説明がリストで表示されます。リストを選択してダブルクリックして頂くと該当するソース行にジャンプできます。

ファイル名	行番号	補足説明	
kconv.c	77	ファイルが改行文字で終了していない	
68 69 /* 無変換で標準入力が 70 void copy() 71 { 72 int c1; 73 74 while ((c1 = getc 75 putchar(c1); 76 }	ら標準と char())	↓ 出カヘコピー */ != -1) { /* 1文字標準入力から */ /* 1文字を標準出力へ */	

表示オプション

表示順 - ルール番号順: 解析結果をルール番号順で表示します。

表示順 - ソース位置順: 解析結果をファイル毎に行番号順で表示します。

表示モード- 全体表示: 全ルールのメッセージを表示します.

表示モード - 簡易表示: ルール情報のみ表示し、ファイル名や行番号などソース位置に関する情報は表示されません。

表示モード - 部分表示: 部分表示モードでは、本設定の右側にあるルールメニューで 選択して 頂いたメニューについて表示します。

※上記表示モードの説明は表示順でルール番号順を選択頂いたのみのご説明になります。

ルールの選択

下記ダイアログにて結果表示させるルールを選択する事が可能です。

- 1. CasePlayer2 メニュー「プロジェクト」→「全 MISRA-C 解析」.→「MISRA-C:2004 解析結 果」をダブルクリックして下さい。
- 2. 仕様書ビューア メニュー「ルール」→「表示ルール設定」をクリックして下さい。

ルールの表示設定(M	ISRA-C:2004》	X
インポート(D		別名保存 既定値とする
全ての必要	推奨	[情報] 非举规 非対応 解除
「自動」	車用C言語利用のガイ	イドライン」の連脱時の対応例を付加したルール
全項目	ID 種別	ルール 素示(0)
2. 言語拡張	⊞.∎	すべてのコードは、ISO/IEC 9899:1990, Programming languages - C(ISO/
3. 又書化 4.文字集合	│ ☑ … 必要	未定義又は未規定の動作に依存してはならない。
5.識別子 6.型		言語/コンバイラ/アセンフラが準拠するオフシェクトコードの共通のインタフェー人 コンパイラ/アセンフラが準拠するオフシェクトコードの共通のインタフェー人 コンパイラ/アセンフラが準拠するオフシェクトコードの共通のインタフェー人 の、パイラア
7.定数 8 宣言及び定差		「学動小数点の実装は、定義された浮動小数点規格に従うべきである。
9.初期化 10.管洗刑亦通		アセンブリ言語は、カプセル化して分離しなければならない。
11. ボインタ型の変換	⊞∙☑ 必要	ソースコードでは, /* … */スタイルのコメントだけを用いなければならない。
12.式 13.制御文の式		文字列 /* をコメント内で用いてはならない。
14.制御フロー 15.switch文		コードの一部を「コメントパワト"すべきではない。 田小ふ如理の完美の新佐住 オバアカませし かけりげからかい
16.関数 17. ポインク及び限列	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	市いる処理来と戦の動作は、94CCと著作しては14はならない。 文字集合及びそれに対応するエンコーディングは、文書化しなければならない。
18.構造体及び共用体	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	選定したコンパイラの整数除算の実装について確認し、文書化し、十分に配成
20.標準ライブラリ		用いる#pragma指令は、すべて文書化し、説明しなければならない。
21.実行時誤り	⊞⊡ 必要	ビットフィールドの処理系定義の動作とパッキングに、(プログラムが)依存してい
		製品コードで用いられるすべてのライブラリは、この文書の規定に従って記述され
	·····································	
▼カージルの色	KGB(205, U, U)	【 <u>え史…</u> 】 ■ RGB(0, 0,255) 【 <u>変更…</u> 】
		▼ 非準拠メッセージの文字色 RGB(0, 0, 0) 変更…
	OK	キャンセル ヘルプ(H)
		1

ルールは項目別に分類されています。このリストの項目の選択によって、ルールリストの表示内容が更新されます。チェックを ON にして頂いたルールが解析結果に表示されます。

ソースコード・メトリックス閲覧

プロジェクトビュー(ファイルタブ)の[MISRA-C 解析情報]]→[ソースコード・メトリックス] をダブルクリックして頂くと表示されます。

図 仕様書ビューア - [ソースコード・メトリックス - test.mmtx] □ □ ■ ×												
□ ファイル(F) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)												
🗋 🗅 🚅 🗔	D 🖆 🖬 🕯 X 🕫 🛅 b 🕼 🎆 🎒 🗳 🔗 😵											
表示モード 全	表示モード 全体表示 ▼ 表示順 経路複雑度 ▼ 降順 ▼											
レース	シースコード・メトリックス											
関数名	定義ファイ ル名	行番号	経路複 雑度	マイヤ ーズタ ーバル	ネスト の深さ	コード ・サイ ズ	コメン トを含 む行の 数	コメン ト比率	静的バ スカウ ント			
etos	etos.c	10	13	1	4	40	18	45	85	E		
main	kconv.c	34	11	0	2	34	18	52	90			
stoe	stoe.c	3	9	1	5	36	12	33	27	1		
сору	kconv.c	71	2	0	1	7	2	28	2	1		
f	etos.c	54	1	0	0	3	0	0	1			
f00	f00 etos.c 5 1 0 0 2 0 0 1											
レディ										書禁言		

ソースコード・メトリックスでは下記の情報が出力されます:

関数名: 関数名が表示されます。
定義ファイル名:関数の定義されているソースコード名が表示されます。.
行番号: 関数の定義されているソースコード上での行番号が表示されます。
経路複雑度: 分岐の下図
マイヤーズ・インターバル: 分岐の数+"?:","&&","||"の数が表示できます。
ネストの深さ: 一番深いネスト。
コード・サイズ: 関数の行数 ({から}までの行数)。
コメントを含む行の数: 上記の行のうちコメントを含む行のカウント。.
コメント比率: コメントを含む行のコード・サイズにおける割合。
静的バスカウント: 静的パスをカウントしたもの(総経路数)。

ソースコード・メトリックスの解析結果表示を絞り込んで表示する事が可能です。 仕様書ビューア メニュー「ツール」→「表示設定」をクリックして下さい。

例として、静的パスカウントのチェックを ON、数値を 50 に設定して下さい。 静的パスカウントが 50 以上の関数がソースコード・メトリックスのリストに表示されます。

ソースコード・メトリックス表示設定	X
絞り込み表示条件	OK
	キャンセル ヘルプ(H)
 マイマース・1 ノダーハル ≦ ネストの深さ ≧ 	
 □ード・サイズ ≧ □ □ ×ントを含む行の数 ≧ 0 	
 ▼ 静的パスカウント ≧ 50 □ コメント比率 ≧ 100 	
or コメント比率 ≦ 0	
☑ マイヤーズ・インターバルを経路複雑度との差分にする	

条件設定の絞り込みにより下記のように表示されます::

7	図 仕様書ビューア - [ソースコード・メトリックス - test.mmtx] □ □ ▼										
	□ ファイル(F) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)										
	D 🖨 🖬 🖞 X い 🗐 B B 🕷 🚳 🚇 🧇 🤋 🐼										
₹	表示モード 縦込み表示 → 表示順 経路複雑度 → 降順 →										
	シースコ	ード・メー	トリッ	クス							
Ι.					1		1		1		,
	関数名	定義ファイ ル名	行番号	経路複 雑度	マイヤーズ・	ネスト の深さ	コード	コメン トを含	コメン ト比率	静的バ スカウ	
					インタ ーバル		ズ	む行の 数		レト	
	etos	etos.c	10	13	1	4	40	18	45	85	
	main	kconv∙c	34	11	0	2	34	18	52	90	
-											
107	-1										者宗

これで、MISRA-C解析についての説明を終了します。

7. 検索機能

詳細仕様書ブラウザではプロジェクト内の、関数、変数、typedef、構造体などの様々な情報を検索 する事が可能です。

- 1. 本チュートリアルで使用している test.vproj を開いてください。
- 2. CasePlayer2 メニュー [ツール]→[仕様書ブラウザを開く]をクリックして下さい。
- 3. 種類.を関数に設定して下さい。
- 4. 名前の項目は空白のままにしてください。
- 5. 検索ボタンをクリックして下さい。

	詳細仕様	書ブラウザ	6	-	-			
ः इ	名前 「重類」【関語 定義/宣	<u>故</u> 言	•			★ 検索]	終了(X) ヘルブ(H)
	種別	ソースファイル名	行番号	関数	構造体	種類	限定名	

プロジェクト内にある関数名が表示されます。

検索		×
名前	* 検索	OK キャンセル
	copy etos exit f f00	
	main putchar stoe stricmp	

6. リストの中から getchar を選択して、OK ボタンをクリックして下さい。

類 関数 義 / 宣言			•			検索	ヘルプ
 種別 ソースファ	イル名	行番号	関数	構造体	種類	限定名	
宣言 etosc		14			グローバル関	getchar	
(4)送金の新酒 (1)	-77-74				▼ 開((())	
	~~???170						
111森香V/1里秋 []/* :照/代入	~~//////						
111来音の111秋日 <u>ク</u> 照/代入 絞り込み関数名	~~~					 リスト更新	近 サブ関数もき
111株香い111株 照/代入 線り込み関数名 種別	- XJF1W 	-スファイル4	2	行番号	関数	… リスト更新 構造体	∬ ● サブ関数もき
111株香い111株 照/代入 線り込み関数名 種別 参照	y-	-2771ルส์ รถ	2	行番号 14	関数 etos	リスト更新 構造体	所 🕡 サブ関数もき
111株香い111株 一般 一般 一般 一般 一般 一般 一般 一般 一般 一般	y- etos etos	-スファイルネ sc sc	2	行番号 14 17	関数 etos etos	リスト更新 構造体	所 🔽 サブ関数もき
111株香い111株 一般 一般 一般 一般 一般 一般 一般 一般 一般 一般	y- etos etos	-スファイルネ sc sc sc	2	行番号 14 17 22	関数 etos etos etos	リスト更新 構造体	所
111株香い111株 一 一 一 一 一 で 一 代入 一 で 代入 種 別 参 照	y- etos etos etos	-スファイルネ sc sc sc sc	2	行番号 14 17 22 40	関数 etos etos etos etos	… リスト更新 構造体	所
111株香の111株 一 一 一 一 で 二 11 一 秋 11 11 秋 11 11 秋 11 11 秋 11 11	y- eto: eto: eto: eto: kco	-スファイルส sc sc sc sc sc	2	行番号 14 17 22 40 74	関数 etos etos etos etos copy	… リスト更新 構造体	所 🔽 サブ関数もき
111株香の111株 一 一 一 一 で 照 / 代入 一 種 別 参 照	y- eto: eto: eto: eto: sto: sto:	-スファイルネ sc sc sc sc sc nvc ec	2	行番号 14 17 22 40 74 6	関数 etos etos etos etos copy stoe	… リスト更新 構造体	所 🔽 サブ関数もき
11 11年春001111年 一般 一般 一般 一般 一般 一般 一般 一般 一般 一般	y- eto: eto: eto: eto: sto: sto: sto:	-277114 sc sc sc sc sc sc ec ec	2	行番号 14 17 22 40 74 6 16	関数 etos etos etos etos copy stoe stoe	… リスト更新 構造体	所 🔽 サブ関数もき

getchar 関数が定義/宣言されている位置、また参照/代入されている位置の情報が表示されます。 関連するソースファイルの行を表示するには、リストをダブルクリックして下さい。

12 13	int galji;	
15 16	while ((cl = getchar()) != -1) { /* 1文子標準入力から */ gaiji = 0; /* 外字フラグのクリア */ if (cl == 0x8f) { /* 外字エリアか? */	
17 18 19	if ((c1 = getchar()) == -1) /* 1文字標準入力から */ exit(2); /* 異常終了 */ gaiji = 1; /* 外字フラグセット */	
20 21 22	} if (c1 >= 0xa1) { /* JIS漢字エリアか? */	
23	exit(2);	
25 26 27	$c_2 = c_2 - 0xa1 + 0x40;$ if (c2 > 0x7e) c2++;	
28 29 30	} else c2 = c2-0xa1+0x9f;	-

詳細仕様書ブラウザでは検索したい項目/設定をして頂く事によりプロジェクト内の情報を検索する 事が可能です。

8. 印刷機能

CasePlayer2 では仕様書を実際に印刷する前に. 印刷プレビュー機能を使用して 実際の印刷イメージを確認し、カスタマイズする事ができます。印刷ドキュメントに無駄な空白部分 がない事やページ数が無駄に増えていないか等の確認に役立てて頂けます。

- 1. プロジェクトビュー(ファイルタブ)を表示して下さい。
- 2. [仕様書]→[フローチャート]をクリックして下さい。
- 3. 仕様書リストビュー内に表示される etos をダブルクリックして下さい。
- 4. フローチャートビューア メニュー [ファイル] →[印刷プレビュー]をクリックして下さい。
- 5. メニューバーの下に現れる[2ページ]ボタンをクリックして下さい。



6. [印刷倍率]の右側に位置する (+) or (-)をクリックする事によって、印刷倍率を設定する事が できます。



変更した印刷プレビューの設定は、仕様書を再生成した後も維持されます。

9. 仕様書の Word 変換

仕様書は、CasePlayer2の機能を使用して Word 変換する事ができます。

- 1. CasePlayer2 メニュー「プロジェクト」→「Word 変換」.をクリックして下さい。
- 2. Word 変換する仕様書の種類を右側の[<<変換する仕様書の種類>>]に追加して下さい。
- 3. [OK]ボタンをクリックして下さい。

Word変換の設定 変換する仕様書の選択と順序 <<変換しない仕様書の種類> <u>外部変数一覧</u> typedef一覧 #define一覧 インウルード関係図 外部変数リファレンスリスト 関数リファレンスリスト	>	<<変換する仕様書の種類>> ソースファイル フローチャート モジュール仕様書 構造体仕様書 モジュール構造図【大】 モジュール構造図【小】	
順序	□ ソース毎(こまと) □ プロジェクト	りる(S) 全体を先頭にする(R)	
インデックスファイル名:			
🔲 画像ファイルを埋め込む			
目次のカスタマイズ(M)		印刷属性の統一(丁)…	▼
■ エラーが発生しても変換を続け	5	この設定を既定	定値にする
Word変換	キャンセル	ヘルプ(H)	

etos.v kconv. sample stoe.v test.v Word婆	/mlt: .vmlt: s.vmlt: /mlt: /dlt: 经换終了.
•	
出力	検索結果1 検索結果2
レディ	

- 4. プロジェクトビュー(ファイルタブ)の[外部ファイル]を展開して下さい。
- 5. [Word]をダブルクリックして下さい。

⊡ test
▶ 📄 ソースファイル
● 仕様書
■… 📄 MISRA-C解析情報
□…□□ 外部ファイル

👿 📴 🤊 🗸 🖉 💂 Index.docx [互換モード] - Microsoft Word 🛛 💻 🔍	】 👷 🤊 🗸 🥑 🚽 1.2 etos&0002_vcht.docx [互換モード] 表 💶 💷 💻 🗡
ファイル ホーム 挿入 ページレイアウト 参考資料 差し込み文書 校閲 表示 🏻	? ファイル ホーム 挿入 ページ レ参考資 差し込き 校閲 表示 デザイン レイアウト & (
Image: Web レイアウト Image: Web レイアウト Image: Difference with the second	印刷 全画面 アウトライン レイアウト 国 アウトライン レイアウト 職員 国 レイアウト 職員 国
文書の表示 マクロ マクロ	文書の表示
L D 2 4 6 10 12 14 16 18 20 22 24 26 20 30 32 34 36 30 42 44 48 50 1 7 1 7 1 7 1 1 10 1 1 10 1 10	No. C / L 4 2 4 6 3410 12: 44 16 18 20 22 24 26 28 30 32 34 36 38 30 42 44 16 48 50 7 1 - - - 7 1 - - - 8 2 - 4 6 9 1 - - 9 2 - 4 6 9 2 - 4 6 9 2 - 4 6 9 2 - 4 6 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - - 9 - - -
₂ ● 15 2 15 ~ 12 . · ● 1 → 2 + 5 + 1/1 文字数: 127 ③ □ 即 □ □ = 60% - + + + + + + + + + + + + + + + + + +	* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

目次ページより各種仕様書の word 文書へジャンプする事ができます。 目次の各項目を[Ctrl+クリック] する事で該当文書を表示できます。

注釈:

本機能のご利用には、お使いの PC に Microsoft Word がインストールされている必要があります。 Word 変換操作を実行する前にご使用中の全ての Microsoft Word 文書を閉じて下さい。 変換時に<Template List>エラーが発生した場合、下記の操作にしたがって、Microsoft Word のセ キュリティ設定を変更して下さい。

Microsoft Word	
<template list=""> GaioProject.dot</template>	
ОК	

CasePlayer2 メニュー [ヘルプ]→[ヘルプ]をクリックして CasePlayer2 ヘルプを起動して下さい。 [検索]タブの検索エリアに" Word 変換時のエラー"を入力して[検索開始]ボタンをクリックして下さい。 い。検索結果に"Word 変換時のエラー"が表示されますのでクリックして下さい。 お使いの Word バージョンに合わせて、セキュリティ設定をご変更下さい。

10. 仕様書の HTML 変換

生成した仕様書は、CasePlayer2の機能を使用して HTML 変換する事ができます。

- 1. CasePlayer2 メニュー「プロジェクト」→「HTML 変換」、をクリックして下さい。
- 2. HTML 変換する仕様書の種類を右側の[<<変換する仕様書の種類>>]に追加して下さい。
- 3. [OK]ボタンをクリックして下さい。

HTML変換の設定	×
変換する仕様書の選択 << << < < < <	
 順序 ◎ 名称順(N) □ フロジェクト全体を先頭にする(R) 	
印刷属性の統一(丁) ② 表紙ファイルを指定する(C) 表紙HTMLファイル名(T):	▼ 参照…
画像ファイルの種類: PNG ▼ この設定を既定	値にする
HTML変換 キャンセル ヘルプ(H)	

- 4. プロジェクトビュー(ファイルタブ)の[外部ファイル]を展開して下さい。
- 5. [HTML]をダブルクリックして下さい。

HTM 形式の文書が作成され、各種仕様書がリンクされますのでクリックする事で 該当する仕様書へジャンプする事ができます。

.

and the second second		- 0 - X -		
C:¥SL1¥SL1 0 Manageme	ent¥CP2_エバリューシ: P - C 🥔 test 🛛 🛛	n ★ \$	C:¥SL1¥SL1 0 Managem	nent¥CP2_I/(J1−5: P × Č 🍯 test × 🔐 🛠 🌣
🍦 🤺 🏠 🕶 🖶	• ページ(P) • セーフティ(S) • ツール(0) • ● • ♪	N 🕐 N 😒	🍰 🧳 🏠 🕶 🖬 🖛 🖾 🛊	ಈ ▼ ページ(P)▼ セーフティ(S)▼ ツール(O)▼ 🕢▼ 🤚 🕖 風 😒
test · ソースフィル · フーチャト · マンコール(推進) · 福浩佐林建善 · モジュール(構造)(大) · モジュール(構造)(大) · モジュール(構造)(大) · マジィルルモジュール · ファイル・アレール(構造)(大) · マジュール(構造)(大) · マジュール(構造)(大) · マジュール(構造)(大) · マジュール(市) · 福浩佐一覧	ジャンプする仕様書の種類 ・ジャンプレない。 ロースファイル 0 フローチャ・ モジュール仕様書 0 構造体仕様書 0 モジュール モジュール構造図(ハ) モジュールー覧 0 ファイル内モジュールー覧 0 構造体一覧 test仕様書	ト ル構造図[大]	プローチャート etcs Page 1 マ ・ソースファイル ・モジュール市場での大 ・モジュール市場での大 ・モジュール市場での大 ・モジュール市場での大 ・モジュール市場での大 ・モジュールー場 ・モジュール市場の大 ・モジュール市場での大 ・モジュールー場 ・モジュール市場の大 ・モジュールー場 ・モジュール市場の大 ・モジュールー場 ・モジュールー場 ・ファイルバナジュールー場 ・シTop	ジャンプする仕様書の種類 ○ジャンプレズは、 ○ソースファイル ● フローチャート ○ギジュール仕様書 ○構造体仕様書 ○モジュール構造図(大) ○ギジュール構造図(小) ○モジュールー覧 ○ファイル(パモジュールー覧 ○構造体一覧 フローチャート モジュール名 etcs EUCDrらS/ISIC変換 ■####################################
		🔍 100% 🔻		a 100% 👻

11. 詳細解析モードの詳細設定について

本項目では、詳細解析モードの追加説明と致しまして、詳細解析モードで仕様書生成するための 設定方法をご紹介致します。本チュートリアルにて使用するサンプルプログラムはシンプルな構成 であるため詳細設定は必要ありません。しかし、実際の組込み開発プロジェクトでは、詳細設定が 必要な場合があります。

プリプロセッサの設定

CasePlayer2では解析時にプリプロセッサの設定情報を使用してソースコードの解析を行うため、開発時にクロスコンパイラで設定されているプリプロセッサの設定情報を CasePlayer2 側にも与えてください。

- 1. プロジェクトビュー(設定タブ)を表示して下さい。
- 2. [プリプロセッサ]をダブルクリックして下さい。

プリプロセッサの設定	
 ブリブロセッサの展開結果から仕様書を生成する(P) #if等のブリブロ ブリブロセッサの結果を出力する(1) 	セッサ制御をフローチャートに生成しない
□ #if によるコンパイル非対象部分を制御構造に展開する(E)	
#define(デファイン:名[=数値]) - 道加	展開 注加 ↑ ○ ライブラリ ○ ライブラリ ○ 岡 賢治 ○ マ 市 ○ フントロール パネル ごみ箱 □ □ 178
#undef(デファイン名)	
プレインクルードファイル	1 変更 ↓ 削除
	この設定を既定値にする ル ヘルブ(H)

#define (デファイン名 [=数値])

プリプロセッサに渡すデファインマクロを指定してください。複数指定する場合には、改行で区切っ てください。改行するには、Ctrl+Enterキーを押下してください。[追加]ボタンを押下すると、デファイ ンマクロを追加できます。

#undef (デファイン名)

ソース解析前に、指定したデファイン名を無効にします。

プレインクルードファイル

全てのCソースファイルに予めインクルードさせるファイルのファイル名を指定します。お 使いのクロスコンパイラが暗黙のキーワードや定義を有する場合、これらの解析エラーを回 避するために使用します。 システムインクルードパス/ ユーザーインクルードパス

プリプロセッサに渡すシステムインクルードパス(#include <…>のパス)、ユーザーインクルード パス(#include ~…~のパス)を設定します。パスを追加するには、ストボックスの右側にある「追加」 ボタンをクリックして、[フォルダの選択]ダイアログを表示し、指定するフォルダを選択した状態で [OK]ボタンをクリックしてください。

C オプションパラメータ

この設定は、開発に使用しているクロスコンパイラ独自の方言に対応するためのものです。 CasePlayer2の解析エンジンは、指定した言語仕様に対応していますが、クロスコンパイラ独自の 言語仕様(方言)には対応できません。このため、ソースコードに記述された方言を、解析時にどの ように扱うかを設定します。

- 1. プロジェクトビュー(設定タブ)を表示して下さい。
- 2. [Cオプションパラメータ]をダブルクリックして下さい。

Cオプションパラメータ定義 ユーザーキーワード 置き換えキーワード 非抽出コメント	キーワード(K)	ユーザ 種類(S TYPE	ーキーワード i) 後置記述(W)	
モジュール仕様書・概要説明等の抽 構造体仕様書・概要説明等の抽出定義 #pragma定義	キーワード near far	種類 TYPEATT TYPEATT	後置記述	道加(A) 変更(E) 削除(D)
2個 設定しました				

User Keyword ユーザーキーワード

非 ANSI-C またコンパイラ固有のキーワードはユーザーキーワードにて登録して下さい。 以下に登録方法の例を挙げます。

例1:int near p1;

near は「型修飾子」です。これを型修飾子として認識させるために、near のキーワードに 「TYPEATT」指定します。 キーワード:near、 種類:TYPEATT、 後述記述:[空白] **例2:direct int array[100];**

direct は「記憶クラス」です。これを記憶クラスとして認識させるために、direct のキーワードに「STRAGE」指定します。

キーワード: direct、 種類: STRAGE、 後述記述:[空白]

例3:_asm (" ")

_asm は「インラインアセンブラ記述」です。これをインラインアセンブラ記述として認識させるために、 _asm のキーワードに「ASM」指定します。また、このキーワードに続く記述が_asm の内容の記述で あることを示すために、後置記述に「(EXPRESSION)」を指定します。

キーワード:_asm、 種類:ASM、 後述記述:(EXPRESSION)

例4: _except(.....)

_except は例外処理を示すコンパイラ浦独自の記述文法です。この記述全体を無視させるため に、_except のキーワードに「IGNORE」指定します。また、このキーワードに続く記述が_except の内容の記述であり、同時に無視させるために、後置記述に「(EXPRESSION)」を指定します。 キーワード:_except、 種類:IGNORE、後述記述:(EXPRESSION) 例5:INT32 val;

INT32 は変数の型を表します。型として認識させるために INT32 のキーワードに「TYPE」を指定します。

キーワード:INT32、 種類:TYPE、 後述記述:[空白]

置き換えキーワード

コード内のキーワードを置き換えるために使用します。

例1: typedef __WCHAR_T_TYPE_ _Wchart;

このエラーを回避するために、_WCHAR_T_TYPE_を int 型に置換して解析させます。 新キーワード:_WCHAR_T_TYPE_, 既にあるキーワード int **例2:typedef _SIZE_T_TYPE_ _Sizet;** このエラーを回避するために、_SIZE_T_TYPE_を int 型に置換して解析させます。 新キーワード:_SIZE_T_TYPE_, 既にあるキーワード int

Cオプションパラメータのその他の設定について

大部分のエラーは、上記のユーザーキーワードと置き換えキーワードを使用して回避することがで きます。上記以外には下記の設定がございます。

非抽出コメント モジュール仕様書・概要説明等の抽出定義 構造体仕様書・概要説明等の抽出定義 #pragma 定義

12. 最後に

これにて、CasePlayer2のチュートリアルを終了致します。 本チュートリアルにて、CasePlayer2の操作方法について少しでもご理解を深めて頂ければ幸いです。

CasePlayer2 チュートリアル

※会社名・商品名は各社の商標または登録商標です。 ※本資料の無断転載、複写は禁止しております。

ガイオ・テクノロジー株式会社

■ユーザーサポートのご案内 <u>http://www.gaio.co.jp/support/support_about.html</u>

■使用方法に関するお問い合わせ方法 ご質問は、ユーザーサポート窓口をご利用ください。 <u>http://www.gaio.co.jp/support/support_entry.html</u> ユーザーサポート窓口へのお問い合わせには、ユーザーIDが必要です。 ※保守契約がない場合は、いかなるサポートも提供致しません。