

Universal Probe

ソフトウェア・ユーザーズ・マニュアル

Boundary-Scan Tool

Copyright © 2014 Sohwa & Sophia Technologies Inc.

No. J090963-02

目次

注意事項 9

使用上の注意 10

略語・用語・記載ルール 12**1. 本ソフトウェアの概要 14**

1.1. システム要件	14
1.2. 特徴	14
1.3. 電源の入れ方/切り方	15
1.3.1. ターゲットとの接続方法と電源の入れ方	15
1.3.2. 電源の切り方とターゲットからの外し方	15

2. 操作方法 16

2.1. 作業フロー	16
2.1.1. Boundary-Scan Tool 起動	17
2.1.2. プロジェクトビューの操作手順	18
2.1.3. 波形表示ウィンドウの操作手順	19
2.1.4. 端子状態一覧ウィンドウの操作手順	19
2.1.5. 接続テストウィンドウの操作手順	20
2.1.6. スクリプトペインの操作手順	20
2.2. プロジェクトビュー	21
2.2.1. プロジェクトの新規作成	21
2.2.2. プロジェクトを開く	21
2.2.3. プロジェクトの保存	22
2.2.4. プロジェクトを閉じる	22
2.2.5. デバイスの追加	24
2.2.6. デバイスの編集	25
2.2.7. デバイスの削除	26
2.2.8. デバイスの上移動	26
2.2.9. デバイスの下移動	26
2.2.10. デバイスのプロパティ	27
2.2.11. プローブの接続	29
2.2.12. プローブの選択	29
2.2.13. プローブの通信（デバイスの自動認識）	31
2.2.14. プローブの切断	33
2.2.15. JTAG クロックの設定	34
2.2.16. JTAG クロックの選択	34

2.2.17. JTAG クロックの設定（低速）	35
2.2.18. 整合性テスト	36
2.2.19. ポートグループの追加	37
2.2.20. ポートグループの編集	38
2.2.21. ポートグループの削除	38
2.2.22. デバイス参照名の設定	39
2.2.23. デバイス参照名のリセット	40
2.2.24. ポート名の変更（デバイス単位）	40
2.2.25. ポート名の変更（ポート単位）	41
2.2.26. ポート名のリセット（デバイス単位）	41
2.2.27. ポート名のリセット（ポート単位）	41
2.2.28. ポート名変更の保存	41
2.3. 波形表示ウンドウ	43
2.3.1. 波形ウンドウの新規作成	43
2.3.2. 波形をビューワで開く	43
2.3.3. 波形の名前変更	44
2.3.4. 波形の削除	45
2.3.5. ポート（信号）の追加	46
2.3.6. ポートグループ（信号）の作成	47
2.3.7. ポートグループ（信号）の2進/16進表示 切り替え	48
2.3.8. ポート/ポートグループ（信号）の表示順序変更	48
2.3.9. ポート/ポートグループ（信号）の削除	48
2.3.10. サンプリング周期の選択	49
2.3.11. サンプリング開始（計測開始）	50
2.3.12. サンプリング停止（計測停止）	50
2.3.13. ズームイン	50
2.3.14. ズームアウト	50
2.3.15. カーソルの設定	51
2.3.16. カーソルの移動	51
2.3.17. カーソル位置を中央に表示	51
2.3.18. マーカの設定	52
2.3.19. マーカ位置へカーソルを移動	53
2.3.20. マーカ位置を中央に表示	53
2.3.21. 波形の印刷	54
2.3.22. 波形の保存	54
2.4. 端子状態一覧ウンドウ	55
2.4.1. 端子状態一覧の新規作成/開く	55
2.4.2. リストの名前変更	55
2.4.3. リストの削除	56
2.4.4. ポートの追加	57
2.4.5. ポートグループの作成	58

2.4.6. ポートグループの 2 進/16 進表示切り換え	59
2.4.7. ポート/ポートグループの表示順序変更	59
2.4.8. ポート/ポートグループの削除	59
2.4.9. SAMPLE コマンド実行	61
2.4.10. EXTEST コマンド実行	61
2.4.11. CSV ファイルに保存	64
2.5. 接続テストウィンドウ	65
2.5.1. 接続テストウィンドウの新規作成/開く	66
2.5.2. テスト（リスト）の名前変更	66
2.5.3. テスト（リスト）の削除	67
2.5.4. ネットの追加	68
2.5.5. ピンリストの編集	69
2.5.6. ネットの削除	70
2.5.7. ネットの表示順序変更	70
2.5.8. ネットリストを開く	71
2.5.9. グループの作成/編集/削除	71
2.5.10. ピン/ポート名表示切り替え	73
2.5.11. 出力値の設定	73
2.5.12. 期待値の設定	74
2.5.13. テストの実行	75
2.5.14. 2 進/16 進表示切り替え	76
2.5.15. ネットリストの保存	77
2.5.16. 接続テスト結果の保存	78
2.6. スクリプトペイン	79
2.6.1. ファイル選択	80
2.6.2. 実行（ファイル）	81
2.6.3. 実行（コマンドライン）	82
2.6.4. 保存（テキスト形式）	83
2.6.5. 保存（スクリプト形式）	83
2.6.6. 表示のクリア	83

3. メニュー **84**

3.1. ファイル	84
3.1.1. プロジェクトの新規作成	84
3.1.2. プロジェクトを開く	84
3.1.3. プロジェクトの保存	85
3.1.4. プロジェクトを閉じる	85
3.1.5. 波形ウィンドウの新規作成	87
3.1.6. 波形の保存	87
3.1.7. 波形をビューワで開く	87
3.1.8. 端子状態一覧の新規作成/開く	87

3.1.9. 端子状態一覧を CSV 形式で保存	88
3.1.10. 接続テストウィンドウの新規作成	88
3.1.11. ネットリストを開く.....	88
3.1.12. ネットリストの保存	88
3.1.13. 接続テスト結果の保存	88
3.1.14. 印刷プレビュー	88
3.1.15. プリンターの設定.....	89
3.1.16. 印刷	89
3.1.17. アプリケーションの終了	90
3.2. 表示	91
3.2.1. ツールバーとドッキングウィンドウ	91
3.2.2. ステータスバー	92
3.2.3. スクリプト	92
3.3. プロジェクト.....	93
3.3.1. 接続	93
3.3.2. 切断	93
3.3.3. デバイスの追加.....	93
3.3.4. 整合性テスト	93
3.3.5. スクリプトファイルを追加.....	93
3.3.6. ポートグループを追加	94
3.3.7. JTAG クロックの設定.....	94
3.3.8. JTAG 信号の強制出力	94
3.4. 波形表示.....	95
3.4.1. サンプリング開始（計測開始）	95
3.4.2. サンプリング停止（計測停止）	95
3.5. 端子状態一覧.....	96
3.5.1. SAMPLE コマンド	96
3.5.2. EXTEST コマンド実行.....	96
3.5.3. EXTEST 実行時に警告を表示	96
3.6. 接続テスト	97
3.6.1. 期待値の学習	97
3.6.2. 接続テスト.....	97
3.6.3. 信号出力時に警告を表示	97
3.7. ツール	98
3.7.1. BSDL 管理	98
3.7.2. プローブのライセンス取得	99
3.7.3. エディタ指定	100
3.8. ヘルプ	101
3.8.1. ヘルプの表示	101
3.8.2. バージョン情報	101

4. ウィンドウ 102

4.1. プロジェクトビューウィンドウ	102
4.1.1. ツールバー	103
4.1.2. プロジェクト名.....	103
4.1.3. プローブ情報	103
4.1.4. 波形表示ツリー	104
4.1.5. 端子状態一覧ツリー	104
4.1.6. 接続テストツリー	104
4.1.7. スクリプトツリー.....	105
4.1.8. ポートグループツリー	105
4.1.9. ボードツリー	105
4.2. 波形表示ウィンドウ	107
4.2.1. ポートグループ名表示欄	107
4.2.2. カーソル（緑の実線）	107
4.2.3. マーカ（青・橙の破線）	107
4.3. 端子状態一覧ウィンドウ.....	108
4.3.1. ツールバー	108
4.3.2. チェックボックスカラム.....	108
4.3.3. ピンカラム	109
4.3.4. デバイスポート名カラム.....	109
4.3.5. タイプカラム	109
4.3.6. I/O カラム.....	109
4.3.7. 出力カラム.....	110
4.3.8. 入力カラム.....	110
4.3.9. C-Cell（コントロールセル）カラム	110
4.4. 接続テストウィンドウ	111
4.4.1. ツールバー	111
4.4.2. Net 番号カラム.....	111
4.4.3. 出力ピンリストカラム	111
4.4.4. 入力ピンリストカラム	112
4.4.5. 出力値カラム	112
4.4.6. 期待値カラム	112
4.4.7. 入力値カラム	112
4.4.8. 結果カラム	112
4.5. デバイス編集/BSDL 管理ダイアログ	113
4.5.1. メーカーリスト	113
4.5.2. デバイスリスト	113
4.5.3. BSDL ファイルエディット	114
4.5.4. OK・キャンセル	114
4.6. JTAG クロック設定ダイアログ	115

4.6.1. クロック選択	115
4.6.2. 低速クロックエディット	116
4.7. スクリプトペインウインドウ	117
4.7.1. ツールバー	117
4.7.2. ファイル名エディットボックス	117
4.7.3. 標準入出力エディットボックス	118
4.8. ステータスバー	119
5. スクリプト.....	120
5.1. インターフェース関数一覧.....	120
5.1.1. デバイス設定関連	120
5.1.2. 値設定関連	122
5.1.3. 接続テスト設定関連	124
5.1.4. EXTEST 設定関連	128
5.1.5. バウンダリスキャンテスト実行関連	129
5.1.6. 波形表示ウインドウ操作関連.....	130
5.1.7. 端子状態一覧ウインドウ操作関連.....	132
5.1.8. 接続テストウインドウ関連.....	134
5.1.9. 環境設定関連.....	139
5.1.10. その他.....	140
5.2. インターフェース関数使用時の注意	141
5.2.1. 関数の呼び出し方について	141
5.2.2. ピン名の指定について	141
5.2.3. ピンの値について（データフォーマット）	141
5.2.4. ピンの値について（タイプ別の設定/取得）	142
5.2.5. ピンの値について（ドライブ・更新のタイミング）	142
5.2.6. デバイス名について	143
5.2.7. クロック設定について	143
A. 付録	144
A.1. 各種デバイス接続 対応方法	144
A.1.1. Vref 信号が接続されていないターゲット	144
A.1.2. BSDL 情報がないデバイスが認識される場合	147
A.2. メッセージ一覧	151
A.2.1. [ERROR]	151
A.2.2. [WARNING]	152
A.3. トラブルシューティング	154
A.3.1. インストール関係	154
A.3.2. プロジェクトビュー使用時	154
改訂履歴	155

注意事項

このたびは株式会社 Sohwa & Sophia Technologies 製「Universal Probe」をお買い上げいただき、誠にありがとうございます。本書に記載されている注意事項などを正しくご理解のうえ、お使いいただきますようお願い申し上げます。

1. 本書に記載の製品及び技術で、『外国為替及び外国貿易法』に該当するものを輸出する時、又は、国外に持ち出す時は、日本政府の許可が必要です。
2. 本書に記載されている製品は、一般電子機器(事務機器、通信機器、計測機器、家電製品など)に使用されることを意図しております。特別な品質、信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼしたりする恐れのある特定用途機器(自動車・鉄道・船舶・航空・宇宙用機器、交通機器、燃焼機器、安全装置、医療機器、インフラ機器、原子力など)には使用しないでください。もしこれらの機器でご使用になる場合は、お客様の責任のもとでご使用ください。
3. 本書の内容の一部または全部を当社の文書による承諾なしに、無断で転載することは固くお断りいたします。
4. 本書に記載の内容は、将来予告なしに変更される場合があります。
5. 本書に記載の仕様は、お客様の環境、測定条件によって異なる結果が得られる場合があります。
6. 運用した結果の影響について、一切の責任を負いかねますのでご了承ください。
7. 本書に記載の「使用上のご注意」は、使用者や他者への危害と財産の損害を未然に防ぎ、安全に正しくお使いいただくための重要な注意事項です。ご使用になる前に必ずお読みください。
8. 本書に記載されている製品名および商品名は、各社の商標または登録商標です。



連絡先は 株式会社 Sohwa & Sophia Technologies のホームページをご確認ください。
URL > <http://www.ss-technologies.co.jp>

使用上の注意



下記の注意を守らないと人が死亡する、または重傷を負う可能性があります。



強制

本製品に仕様で規定した範囲外の電源電圧を加えないでください。
範囲外での電源電圧を加えると、破損・火災の恐れがあります。



強制

アース端子が付いているターゲットに使用する場合は、ターゲットや周辺機器のアースを確実に接続してください。機器の故障や感電の恐れがあります。
また、ガス管にアース端子をつながないでください。火災や爆発の原因になります。



禁止

本製品に接続した機器を取り付けたまま持ち運ばないでください。
特にケーブルはプラグを持って抜き差してください。ケーブルが破損し、火災・感電の恐れがあります。



禁止

ケーブルを取り扱う場合は次の点を守ってください。「傷つけない」「加工しない」「無理に曲げない」「ねじらない」「引く張らない」「物を載せない」「加熱しない」「熱器具に近づけない」「濡れた手で触らない」。
これらを守らないと火災・感電の恐れがあります。
もしケーブルが破損した場合、そのケーブルの使用を中止してください。



禁止

雷が鳴りだしたら、電源プラグに触れないでください。感電の原因となります。
落雷により製品が破損したと思われる場合は、本製品の使用を中止してください。



禁止

ステープラの針、クリップなどの金属を内部に入れないでください。火災・故障の恐れがあります。



禁止

直射日光の当たる場所、熱器具の近く、極端な高温環境、極端な低温環境、振動の激しいところ、金属や油を含むほごりの多い場所、スパイク系のノイズが発生する場所で使用したり、放置しないでください。
また、強い衝撃を与えないで下さい。



分解禁止

分解・改造・修理しないでください。火災・感電の恐れがあります。



水濡れ禁止

風呂場やコップの近くなど、液体のある場所、湿気の多い場所では使用しないでください。
感電する恐れがあります。

液体が本製品内部に入った場合はすぐに電源を切り、使用を中止してください。



注意

通電中の本製品に長時間触れていると低温やけどになる恐れがあります。
また、本製品を布団などで覆った状態で使用しないでください。



プラグを抜く

もし、異常なにおい・異常な音・発煙・発火した場合、または落としたり、強い衝撃を与えたとして破損、破損した恐れのある場合は、すぐに電源を切ってください。そのまま使うと重大な事故を起こす可能性がありますので、使用を中止してください。



- 本ソフトウェアは各デバイスマーカーより公開されている BSDL ファイルを情報源として動作しています。BSDL に記述されている内容が正しいという前提でデバイスの認識やバウンダリスキャンテストを行うことにご注意ください。
- 本ソフトウェアは弊社が 2011 年 10 月に取得した BSDL ファイルを使用しています。BSDL ファイルは最新のものを使用してください。
- BSDL ファイルが公開されていないデバイスについては動作対象外となります。
- バウンダリスキャンテスト実行時の設定によってはデバイスに過負荷がかかり、**デバイスを損傷する恐れがあります**。テストを行う際は充分にご注意ください。
なお、設定に問題がありデバイスが損傷した場合、弊社は一切の責任を負いかねますのでご了承ください。

略語・用語・記載ルール

本書で使用する略語・用語や記載ルールについて説明します。

- 数値について … 特に記載がない限り、数値はすべてプラスの値とします。
- K(大文字) … $2^{10}=1024$ を表します。(例 : 16K=16384)
- k(小文字) … 1000 を表します。(例 : 1kHz=1000Hz)
- [xxxxx] … xxxx というウィンドウタイトルを示します。

本書で使用する注釈・注意点などについては Figure 1 の通りです。

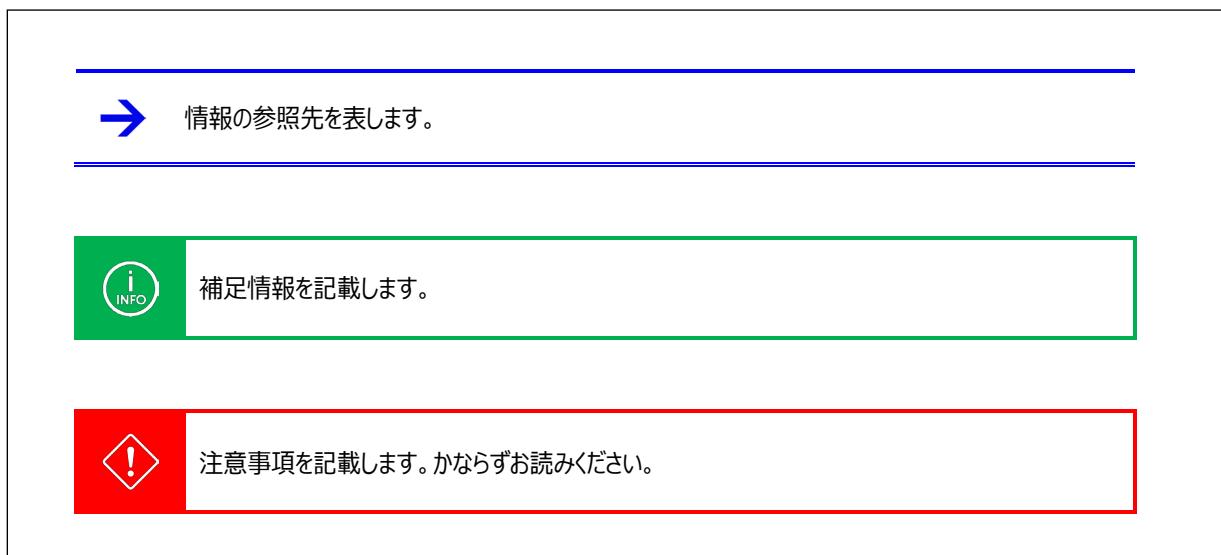


Figure 1

略語・用語の解説は Table 1 の通りです。

Table 1

略語・用語	説明
本製品	Universal Probe 本体・付属品を含むもの。
プローブ	Universal Probe 本体のこと。
本ソフトウェア	Boundary-Scan Tool のこと。
Serial No.	Serial Number の略。プローブの ID のこと。
Software Code	各ソフトウェアの License Code を発行するために必要なライセンス。無償版は不要。
License Code	本機で使用できる機能を追加するためのコード。
ホスト PC	Universal Probe を制御する PC のこと。
ターゲット	Universal Probe によって制御、計測する対象のこと。
(N/A)	Not Assigned の略。割り当てられた情報がないことを表す。
TCK	Test Clock の略。JTAG インターフェースのクロック。
TMS	Test Mode Select の略。接続先の JTAG TAP コントローラのステートを制御する。
TDI	Test Data In の略。JTAG インターフェースのデータ入力。本機からは出力となる。
TDO	Test Data Out の略。JTAG インターフェースのデータ出力。本機では入力となる。
TRST(nTRST)	Test Reset の略。JTAG TAP コントローラをリセットする信号。(オプション)
SRST(nSRST)	System Reset の略。通常はターゲットのメインリセットに接続する。(オプション)
RTCK	接続先から出力されるクロック。デバッグプローブは TCK をこの周波数に合わせる。(オプション)
DBGREQ	Debug Request の略。デバッグプローブからターゲットを停止してデバッグ状態にする。(オプション)
DBGACK	Debug Acknowledge の略。DBGREQ によってターゲットが停止すると変化する。(オプション)
BSDL ファイル	Boundary Scan Description Language の略。通常はデバイスマーカーから入手する。

1. 本ソフトウェアの概要

1.1. システム要件

- Microsoft Windows 7 以降が動作する PC
- CPU : 1GHz 以上 (使用する OS の要件に準拠します)
- Memory : 1GB 以上 (使用する OS の要件に準拠します)
- HDD : 空き容量 500MB 以上
- OS : Windows 7 以降 (32bit または 64bit)
- USB 2.0 の空きポート 1 つ以上

1.2. 特徴

Boundary-Scan Tool は、Windows 7 以降で動作する JTAG 機構を活用したデバイス端子の状態確認/操作するためのツールです。

Boundary-Scan Tool は、以下の特徴/機能があります。

● 波形表示

- 指定した端子状態の推移を波形表示します。
- 最小 10ms 間隔で信号をサンプリングすることができます。
- 測定した波形をファイルに保存しておくことができます。保存した波形を波形ビューワで見ることができます。
- 測定した波形を印刷することができます。
- 各設定をファイルに保存しておくことができます。

● 端子状態一覧

- バウンダリスキヤンテストの SAMPLE 命令により指定した端子の状態を確認することができます。
- バウンダリスキヤンテストの EXTEST 命令により端子間の繋がりをテストすることができます。
- 表示している端子状態を CSV ファイルに保存することができます。
- 各設定をファイルに保存しておくことができます。

● 接続テスト

- 指定した端子状態の推移を波形表示します。
- バウンダリスキヤンテストの EXTEST 命令により端子間の繋がりをテストすることができます。
- 端子状態一覧よりも端子間の繋がりに特化した機能です。
- 端子の接続の設定をファイルに保存することができます。
- 接続テストのために出力する値やそのときの期待値をファイルに保存しておくことができます。

● スクリプト実行

- 指定した端子状態の推移を波形表示します。
- Python スクリプトとインターフェース関数を使用して本ソフトウェアの各種機能を実行することができます。
- スクリプトはコマンドラインからの実行とスクリプトファイルからのバッチ実行ができます。
- 実行結果を 2 種類のログファイルに保存することができます。(テキストファイル及び Python スクリプトファイル)
- 使用したスクリプトファイルの情報をファイルに保存しておくことができます。

1.3. 電源の入れ方/切り方

1.3.1. ターゲットとの接続方法と電源の入れ方

以下の手順でプローブとターゲットを接続します。

- 1) ターゲットの電源が切れている事を確認します。
- 2) プローブの USB ケーブルを接続します。
- 3) ターゲットとプローブを接続します。
- 4) ターゲットの電源を入れます。
- 5) 本ソフトウェアを起動し、データの書き込み作業などを行います。

1.3.2. 電源の切り方とターゲットからの外し方

以下の手順でプローブとターゲットを切り離します。

- 1) 本ソフトウェアを終了します。
- 2) ターゲットの電源を切ります。
- 3) ターゲットからプローブを外します。



プローブのハードウェア仕様、ターゲットの制限、接続などの詳細については、『ハードウェア・ユーザーズ・マニュアル』を参照してください。

2. 操作方法

2.1. 作業フロー

Boundary-Scan Tool を起動してから、ユーザーのモジュールを書き込み、プロジェクトファイルを保存するまでの、主な工程を以下に示します。

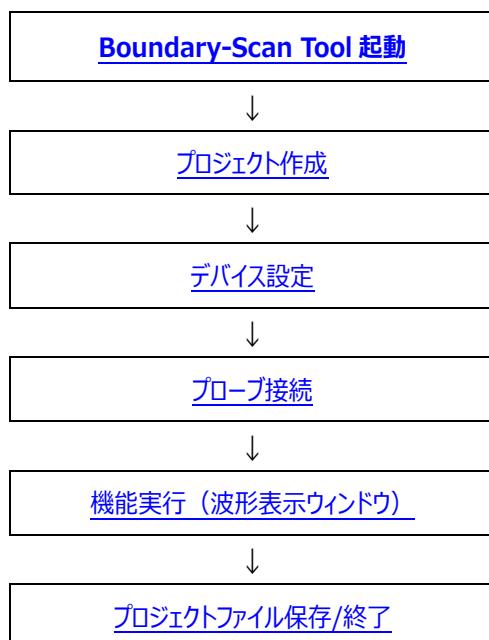
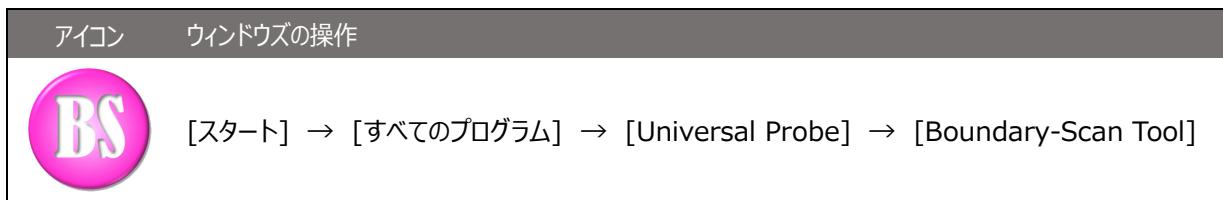


Figure 2



機能実行には、『プロジェクトビュー』、『波形表示ウィンドウ』、『端子状態一覧ウィンドウ』、
『接続テストウィンドウ』『スクリプトペイン』などがあります。
上記フローでは例として、波形表示ウィンドウを示しています。

2.1.1. Boundary-Scan Tool 起動



本ソフトウェアを起動直後は、下図のようなウィンドウが開きます。

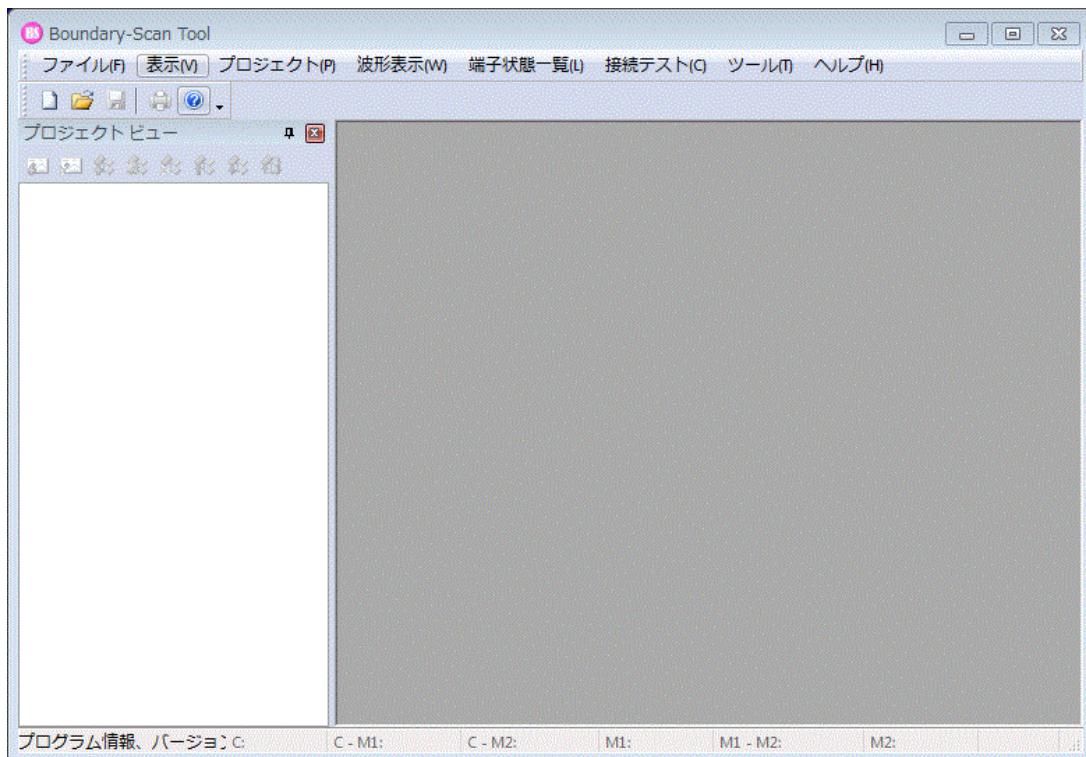


Figure 3



各機能の詳細な操作手順については、[『プロジェクトビュー』](#)、[『波形表示ウィンドウ』](#)、[『端子状態一覧](#)、[『接続テストウィンドウ』](#)、[『スクリプトペイン』](#)の項を参照願います。

2.1.2. プロジェクトビューの操作手順

プロジェクトビューの主な操作手順は、以下のようになります。

Table 2

	①プロジェクト新規作成/開く 空のプロジェクトを作成して各種設定を行います。もしくは既に設定済みのプロジェクトを開きます。
	(②デバイスの手動設定（任意）) デバイスの追加・削除・上移動・下移動・プロパティを用いて JTAG チェーンの構成を行います。
	③プローブの接続（デバイス自動認識） プローブを起動させ、操作可能状態にします。同時にデバイスの自動認識を試みます。
	(④デバイスの手動設定（任意）) 自動認識に失敗、BSDL 情報が間違えている、等があった場合は JTAG チェーンの構成を行います。
	⑤JTAG クロックの設定 バウンドリスキヤンテストを行うためのクロック周波数を設定します。
	⑥整合性テスト プローブ接続後に JTAG チェーンを変更した場合、構成したチェーンが正しいかどうかテストを行います。
	⑦ポートのグルーピング（任意） 複数のポートをグルーピングします。
	⑧他のウィンドウを開く 波形表示ウィンドウ・端子状態一覧ウィンドウを開いてバウンドリスキヤンテストを実行します。 波形表示ウィンドウ 端子状態一覧ウィンドウ
	⑨プローブの切断 プローブとの接続を切れます。
	⑩プロジェクトの保存/閉じる 各種設定をプロジェクトに保存します。テストを終了したプロジェクトを閉じます。

2.1.3. 波形表示ウィンドウの操作手順

波形表示ウィンドウの主な操作手順は、以下のようになります。

Table 3

	<p>①波形表示ウィンドウの表示 波形表示パターンを新規に作成するか、または既存の波形表示パターンを選択して波形表示ウィンドウを開きます。保存しておいた波形を開いて表示することもできます。</p>
	<p>②計測条件を設定 測定を開始する前に、以下に示すサンプリング条件を設定します。 ポートグループの操作 サンプリング周期の選択</p>
	<p>③波形計測 指定された周期でサンプリングし、波形を表示します。</p>
	<p>④波形を検証 サンプリングを停止し、または、保存しておいた波形を開いて検証します。</p>
	<p>⑤波形を保存 後で検証するために、波形をファイルに保存します。</p>

2.1.4. 端子状態一覧ウィンドウの操作手順

端子状態一覧ウィンドウの主な操作手順は、以下のようになります。

Table 4

	<p>①端子状態一覧ウィンドウを開く 空の端子状態一覧リストを作成して各種設定を行います。もしくは既に設定済みのリストを開きます。</p>
	<p>②ポートグループの設定 ポートやグループの追加・削除・上移動・下移動を用いて一覧表示するポートの構成を行います。</p>
	<p>③バウンダリスキヤンテスト実行 SAMPLE コマンド・EXTEST コマンドを使い、バウンダリスキヤンテストを実行します。</p>
	<p>④現在の状態を CSV ファイルに保存する 端子状態一覧ウィンドウに表示されている現在の状態を CSV ファイルに保存します。</p>

2.1.5. 接続テストウィンドウの操作手順

接続テストウィンドウの主な操作手順は、以下のようになります。

Table 5

	①接続テストウィンドウを開く 空の接続テストリストを作成して各種設定を行います。もしくは既に設定済みのリストを開きます。
	②端子の接続（ネット）の設定 出力側と入力側のポートを設定して「ネットリスト」を構築します。またあらかじめ作成しておいたネットリストファイルからの構築もできます。
	③接続テストを実行 テストパターンを作成し、バウンダリスキャンテストの EXTEST コマンドを使って接続テストを実行します。
	④接続設定や結果の保存 ネットリストや接続テストの結果をファイルに保存します。

2.1.6. スクリプトペインの操作手順

スクリプトペインの主な操作手順は、以下のようになります。

Table 6

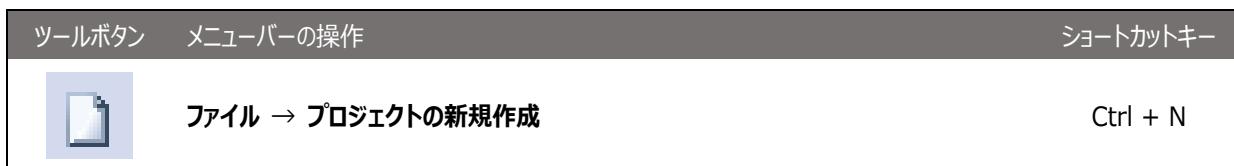
	①スクリプトファイルを選択する バッチ実行する Python スクリプトファイルを選択します。
	②スクリプトファイルを実行する ①で選択したスクリプトファイルを実行して、本ソフトウェアの各種機能をバッチ実行させます。
	③コマンドラインで実行する 標準入出力エディットボックスに Python のコマンドラインを入力して実行させます。
	④履歴の保存・クリア バッチ実行やコマンドライン実行の履歴をテキストファイルや Python スクリプトファイルに保存したり、履歴をクリアしたりします。

2.2. プロジェクトビュー

2.2.1. プロジェクトの新規作成

プロジェクトの新規作成を行います。

初めて使用する場合は、下記ツールボタンもしくはメニューを選択し、プロジェクトを作成してください。



上記メニューを実行すると、プロジェクトビューに「無題」プロジェクトが作成されます。



Figure 4

2.2.2. プロジェクトを開く

以前に作成されたプロジェクトファイルを開きます。前回の作業終了時の設定が復元されます。



上記メニューを実行すると、**ファイルの選択**ダイアログが表示されます。開きたいプロジェクトファイルを選択してから **OK** を選択すると、プロジェクトファイルに記録されている情報がプロジェクトビューに復元されます。

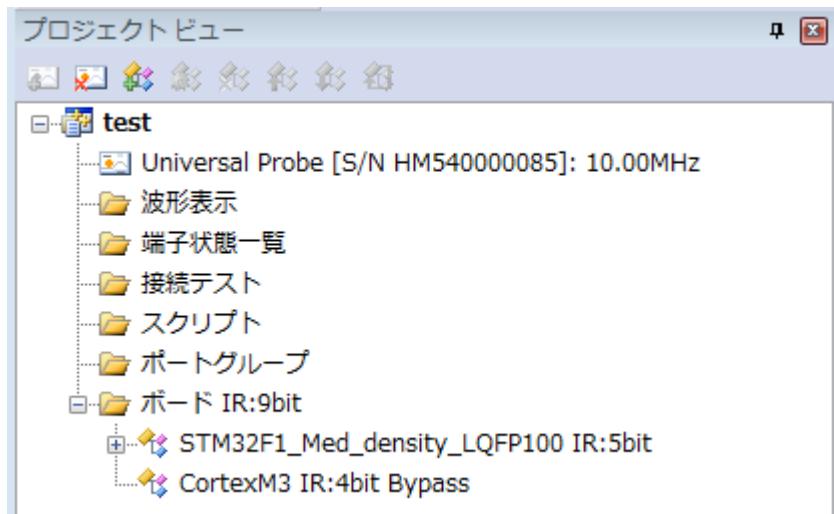
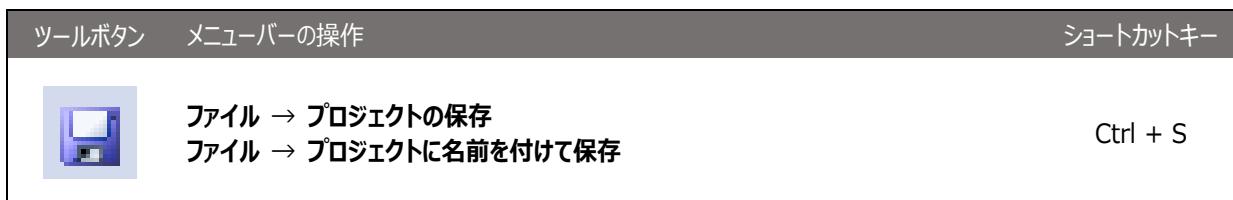


Figure 5

2.2.3. プロジェクトの保存

現在設定されている各種情報をプロジェクトファイルへ保存します。



上記メニューを実行すると、プロジェクトの各種設定をファイルに保存します。保存するプロジェクトのファイルが存在しないときに"上書き保存"を選択するか、"名前を付けて保存"を選択したときはファイルの選択ダイアログが表示されます。保存場所とファイル名を指定してから保存を選択すると、プロジェクトの設定がファイルに保存されます。

プロジェクトファイルの拡張子は .bss です。

プロジェクトファイルは以下の情報を記録しています。

- 接続対象となるプローブの個体 ID とバウンダリスキヤンテストを行うクロック周波数 (JTAG クロック)
- バウンダリスキヤンテストの対象となるターゲットのデバイス構成
- 波形表示ウィンドウに登録したポート情報 (波形表示パターン)
- 端子状態一覧ウィンドウに登録したポート情報 (端子状態一覧パターン) 及び各ポートのチェック状態
- "プロジェクト >> JTAG 信号の強制出力"メニューのチェック状態

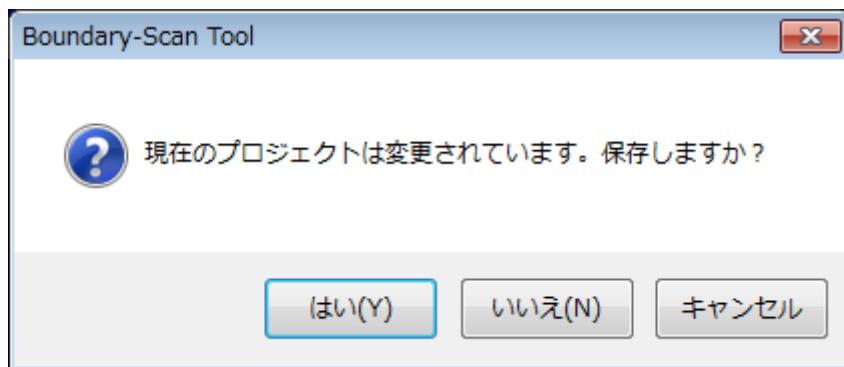
2.2.4. プロジェクトを閉じる

現在開いているプロジェクトを閉じます。

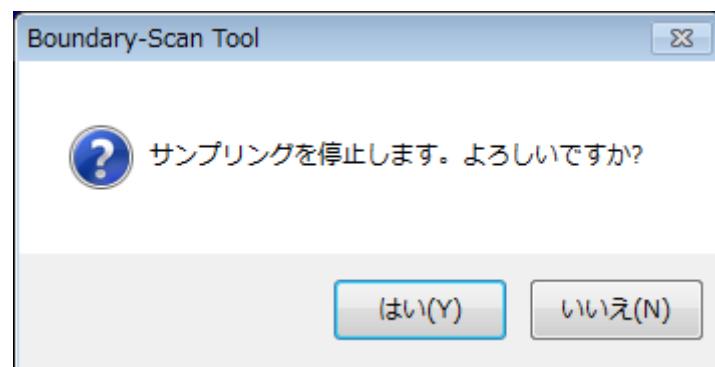


上記メニューを実行すると、現在開いているプロジェクトが閉じプロジェクトビューから表示が消えます。このとき、プローブと接続状態であれば切断します。

プロジェクトが既に作成されていて変更があった状態（プロジェクト名の右に*が付いている）のときに新規作成・開く・閉じるメニューを実行すると、現在開いているプロジェクトを保存するかどうか確認のメッセージが表示されます。「はい」「いいえ」「キャンセル」から目的の動作を選択して先にお進みください。



プロジェクトの新規作成・開く・閉じるメニューの実行時に波形表示ウィンドウがサンプリング中だった場合、**計測を停止するかどうかの確認と波形を保存するかどうかの確認**メッセージが表示されます。「はい」「いいえ」「キャンセル」から目的の動作を選択して先にお進みください。

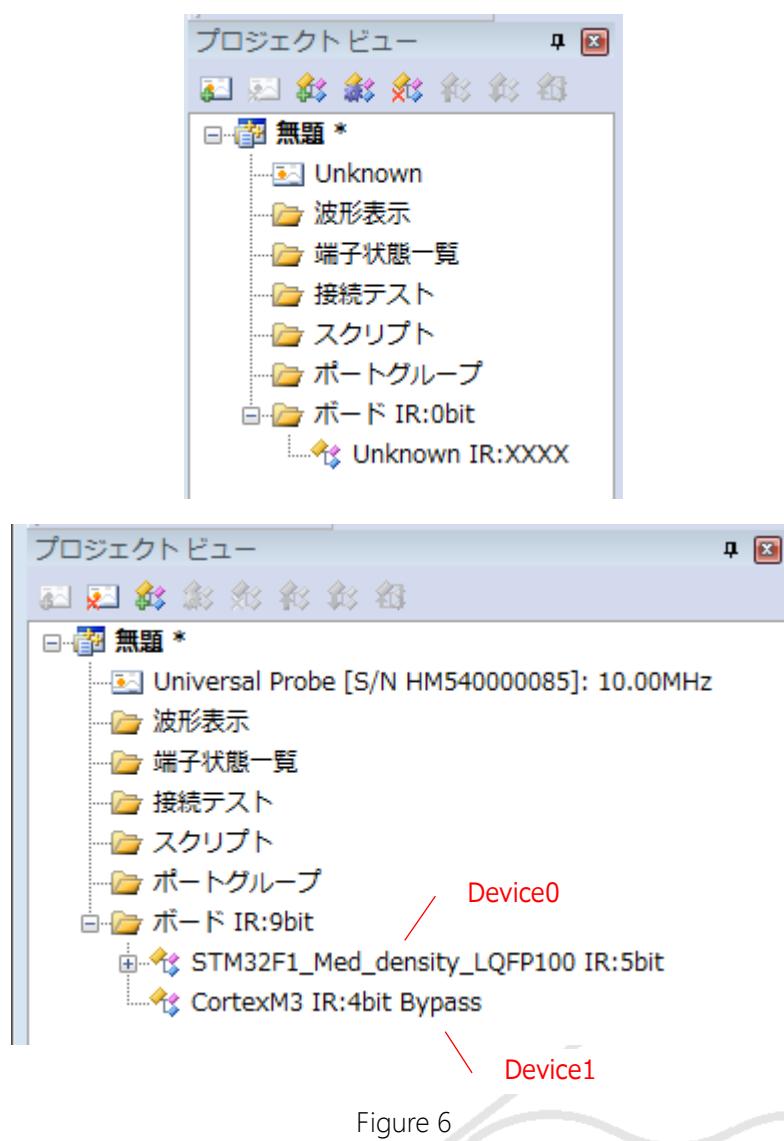


2.2.5. デバイスの追加

プロジェクトにデバイスを設定し、バウンダリスキャンチェーンを構成します。
ターゲットが手元にない、BSDL に間違いがあるなど、何らかの理由で自動認識が出来ない場合に使用します。

ツールボタン	メニューバーの操作	ショートカットキー
	プロジェクト → デバイスの追加 *プロジェクトビュー → ボード 右クリックによるコンテキストメニューでも操作可能	---

プロジェクトビューのポートツリー以下にデバイスを追加します。初期値として **Unknown** デバイスが追加されます。本メニューを実行するたびにボードツリーの最後にデバイスが追加されます。

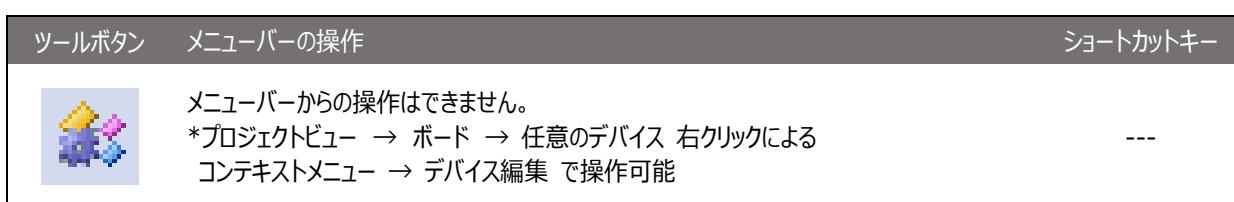


ボードツリー上で一番上にあるデバイスが、プローブの TDI から一番近くにあるデバイスとなります。
また、ボードツリー上で一番上あるデバイスのチェーン番号が常に"0"となり、波形表示ウィンドウや端子状態一覧窗口で"Dev0"として表示されます。



デバイスの番号は、次の図のように TDI に近いデバイスから順になります。
プロジェクトビューにデバイスを登録する際は、TDI に近いデバイスの順で登録して下さい。
プロジェクトビューに表示されているデバイスの順番が実際の接続順番と異なる場合は、「上に移動/下に移動」で、デバイスの位置を合わせて下さい。

2.2.6. デバイスの編集



デバイス編集ダイアログが表示されます。
メーカーリストでメーカーを選択すると本ソフトウェアに登録されているデバイスが一覧表示されますので、お使いのターゲットボードに合わせて適切なデバイスを選択してください。
デバイスが一覧にない場合は、BSDL ファイルを読み込ませることでデバイスを一覧に追加することができます。

BSDL エディットボックスに BSDL ファイルのパスを直接入力するか、 [...]ボタンをクリックして ファイル選択を行ってください。[Read]ボタンを クリックすることで指定した BSDL ファイルが読み込まれ、デバイスが一覧に追加されます。

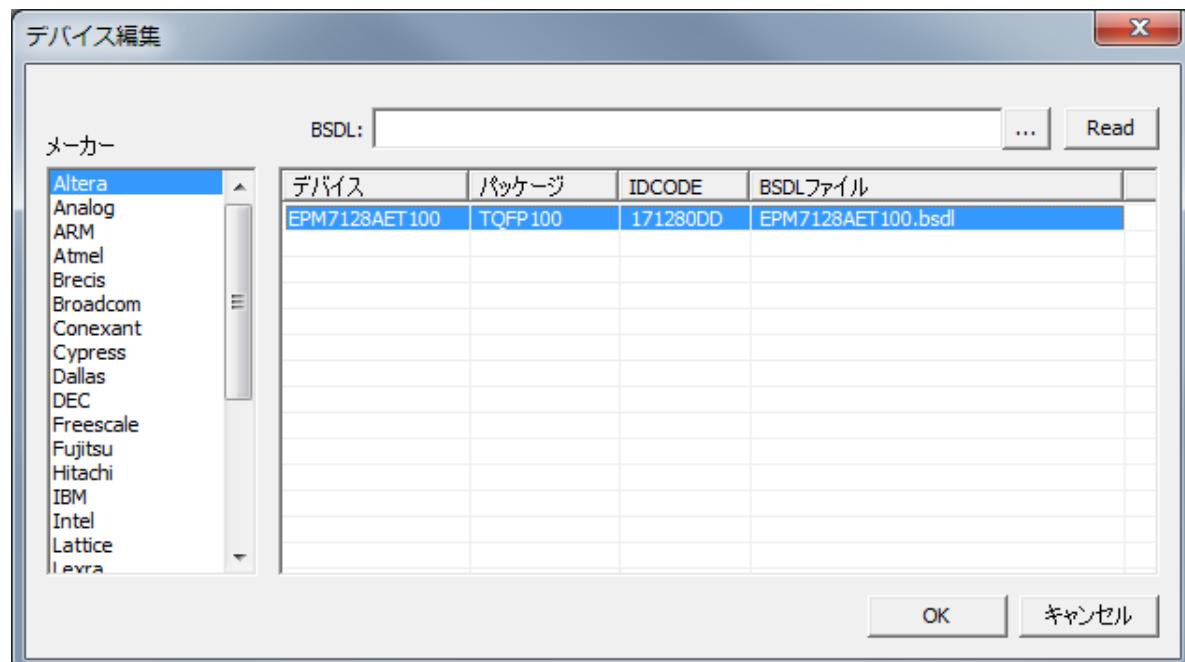


Figure 7

2.2.7. デバイスの削除

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 任意のデバイス 右クリックによる コンテキストメニュー → デバイス削除 で操作可能	---

選択したデバイスがプロジェクトビューのツリーから削除されます。

2.2.8. デバイスの上移動

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 任意のデバイス 右クリックによる コンテキストメニュー → デバイスの上移動 で操作可能	---

選択したデバイスが一つ上のデバイスと入れ替わります。

各デバイスのチェーン番号も入れ替わることにご注意ください。

2.2.9. デバイスの下移動

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 任意のデバイス 右クリックによる コンテキストメニュー → デバイスの下移動 で操作可能	---

選択したデバイスが一つ下のデバイスと入れ替わります。

各デバイスのチェーン番号も入れ替わることにご注意ください。

2.2.10. デバイスのプロパティ

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 任意のデバイス 右クリックによる コンテキストメニュー → デバイスプロパティ で操作可能	---

選択したデバイスのプロパティを表示します。
また、デバイスの IR レジスタのビット長とバイパス設定を編集することができます。

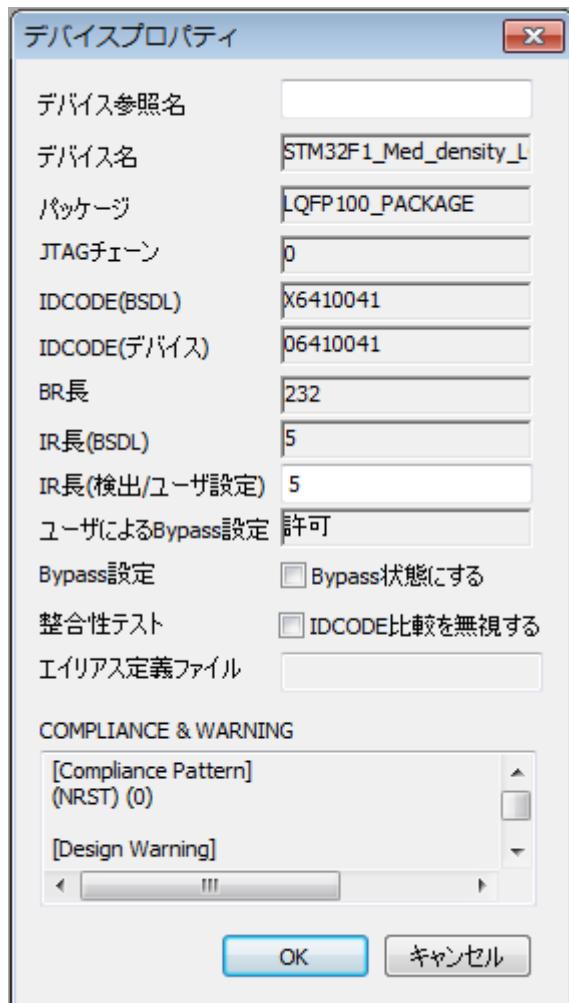


Figure 8

Table 7

デバイス参照名	ユーザーが独自に設定可能なデバイスの名前です。初期値は""になります。
デバイス名	デバイス名称です。初期値は"Unknown"になります。BSDL から取得します。
パッケージ	デバイスのパッケージ情報です。初期値は""になります。BSDL から取得します。
JTAG チェーン	バウンダリスキヤンチェーン上でのデバイスの位置（順番）を表します。 プロジェクトビューのツリー上で、上にあるデバイスほど番号が小さくなります。 最小値は"0"になります。
IDCODE(BSDL)	バウンダリスキヤンの IDCODE 命令により得られる、デバイス固有の ID です。初期値は""になります。 BSDL から取得します。
IDCODE(デバイス)	バウンダリスキヤンの IDCODE 命令により得られる、デバイス固有の ID です。初期値は "00000000"になります。 実デバイスに IDCODE 命令を発行して取得します。 そのデバイスが IDCODE 命令を実装していれば、整合性テストの成功時に更新されます。
BR 長	デバイスのバウンダリスキヤンレジスタのビット長です。初期値は"0"になります。BSDL から取得します。
IR 長(BSDL)	デバイスの IR レジスタのビット長です。初期値は"0"になります。BSDL から取得します。
IR 長 (検出/ユーザ設定)	デバイスの IR レジスタのビット長です。初期値は"0"になります。 自由な値が設定可能です。また、デバイスの自動認識成功時に更新されます。 バウンダリスキヤンのデバイス構成は合っているが整合性テストが通らない場合、ここを変更することでテストが通る可能性があります。
ユーザーによる Bypass 設定	下記の Bypass 設定チェックの有効/無効を表します。BSDL にポート情報が存在しないときに"禁止"になります。 また、"禁止"時にはそのデバイスは強制的に Bypass 状態に設定されます。
Bypass 設定	チェックが付いているとき、そのデバイスはバウンダリスキヤンテスト時に BYPASS 命令によりテストをスキップされます。
整合性テスト	チェックが付いているとき、IDCODE 命令でデバイスから読み出した IDCODE と BSDL の IDCODE による比較を行わずに整合性テストを実施します。BSDL に記述されている内容と IDCODE 命令で読み取れる値との間に食い違いがあるときにチェックを付けてください。
エイリアス定義ファイル	「ポート名の変更」メニューで選択されたファイル名です。
COMPLIANCE & WARNING	BSDL の COMPLIANCE 情報、WARNING 情報が表示されます。 例えば、「ポート N は Low を出力しなければならない」などデバイス固有の決まりごとが表示されます。

2.2.11. プローブの接続



本ソフトウェアと、プローブを接続します。接続時にターゲットのバウンダリスキャンチェーンの自動認識を試みます。

2.2.12. プローブの選択

プローブの接続実行時に PC に繋がれているプローブが 1 台のときはそのプローブと接続します。

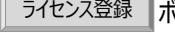
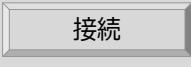
2 台以上のプローブが繋がれているときはプローブ選択ダイアログが表示されるので、使用するプローブを選択してください。



Figure 9

使用するプローブをクリックし、 **接続** ボタンをクリックします。

Table 8

シリアル番号	プローブに登録されているシリアル番号を表示します。
ステータス	プローブの状態を表示します。
	接続可能 : 本ソフトウェアに接続し、使用することができます。 使用中 : 他のアプリケーションで既に使用されています。 本ソフトウェアで使用することはできません。 ライセンス未登録 : ライセンスが登録されていません。  ボタンを押し、ライセンスを登録することで、使用可能になります。
	選択されているプローブに接続します。 ステータスが「接続可能」のプローブを選択時ののみ有効です。
	現在接続されているプローブを再検索し、プローブの一覧を更新します。

ライセンス登録

ライセンスを登録する為のダイアログが表示されます。

ステータスが【接続可能】もしくは「ライセンス未登録」のプローブを選択時のみ有効です。

→ 詳細は『プローブのライセンス取得』の章を参照ください。

閉じる

このダイアログを閉じます。

2.2.13. プローブの通信（デバイスの自動認識）

プローブと接続した後、本ソフトウェアはターゲットのバウンダリスキャンチェーンの自動認識を試みます。ターゲットが接続されていればデバイスの個数を認識したあと、各デバイスの BSDL 情報を取得しようとします。認識したひとつのデバイスに対して複数のパッケージが候補にあるときは、デバイスのパッケージ選択ダイアログが表示されるので適切なパッケージを選択してください。

自動認識に成功するとプロジェクトビューの"ボード"ツリー以下にデバイスが登録されます。

自動認識に失敗した場合は『[デバイスの追加](#)』でバウンダリスキャンチェーンを構成してください。



Figure 10

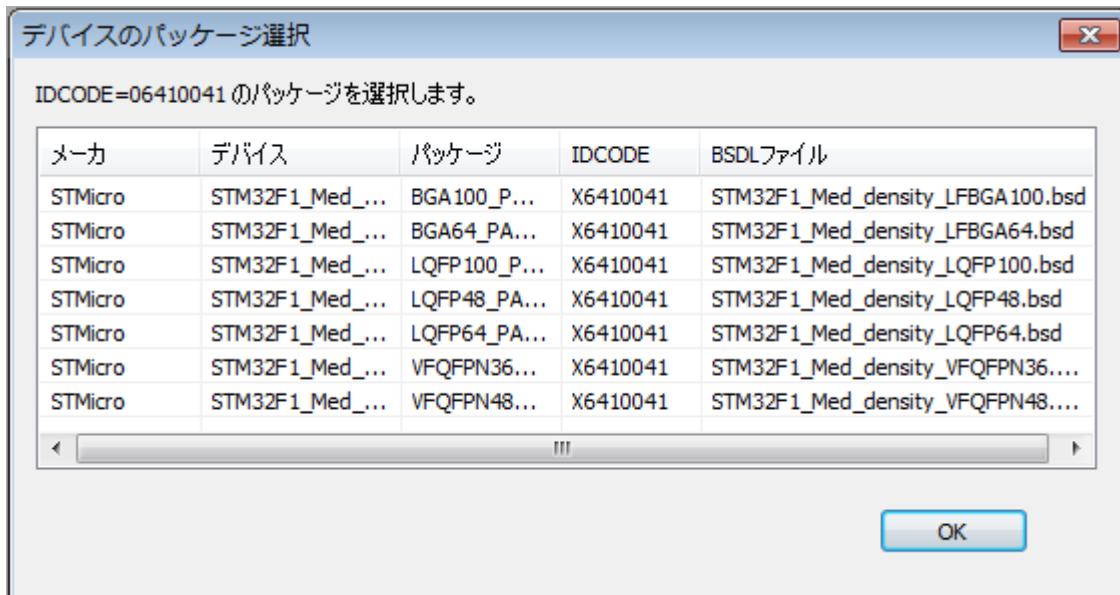


Figure 11

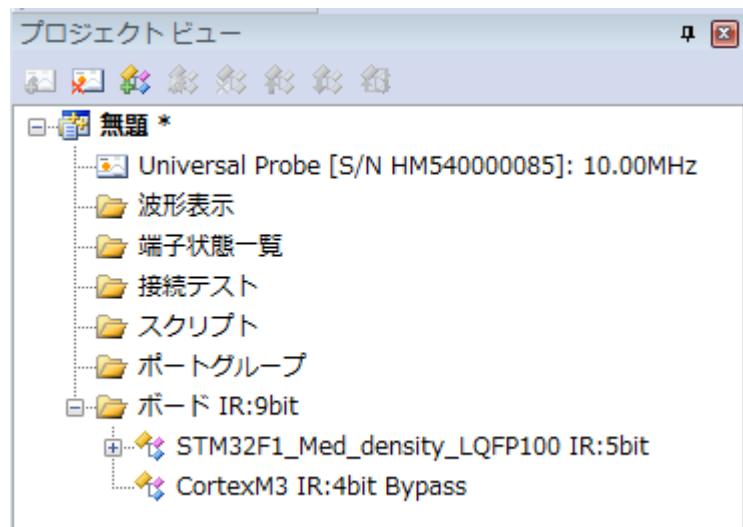
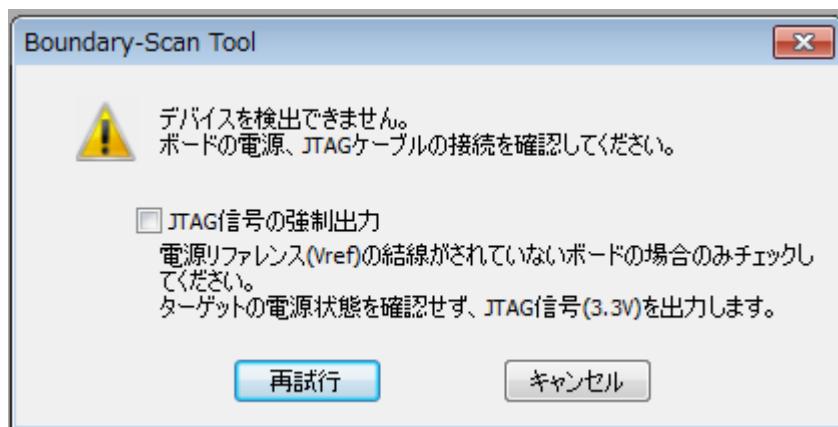


Figure 12



『デバイスの手動設定』や『プロジェクトファイルを開く』であらかじめデバイス構成が出来ている場合は、整合性テストが先に行われます。テストに失敗したときに改めてデバイスの自動認識を試みます。

もしターゲットの電源が検出できずに自動認識に失敗した場合は "デバイスを検出できない" 旨と "ターゲットの電源検出を無視して JTAG 信号を出力するかどうか" を確認する メッセージボックスが表示されます。プローブに電源リファレンスが結線されていないターゲットが繋がっている場合はチェックを付けて再試行をクリックしてください。ターゲットが接続されていない場合はターゲットを繋いで再試行するか、キャンセルボタンから接続処理をキャンセルしてください。



通常は、Vref 信号から JTAG 信号の High レベルとして出力するべき電圧を検出していますが、このチェックを付けた場合、Vref 信号に High レベルとして 3.3V を強制的に出力するようになります。

2.2.14. プローブの切断

ツールボタン	メニューバーの操作	ショートカットキー
	プロジェクト → 切断	Ctrl+Shift+C

本ソフトウェアと、プローブを切断します。その際、プロジェクトからプローブの情報をクリアするかどうか確認のメッセージが表示されます。「はい」「いいえ」「キャンセル」から希望する操作を選んでください。

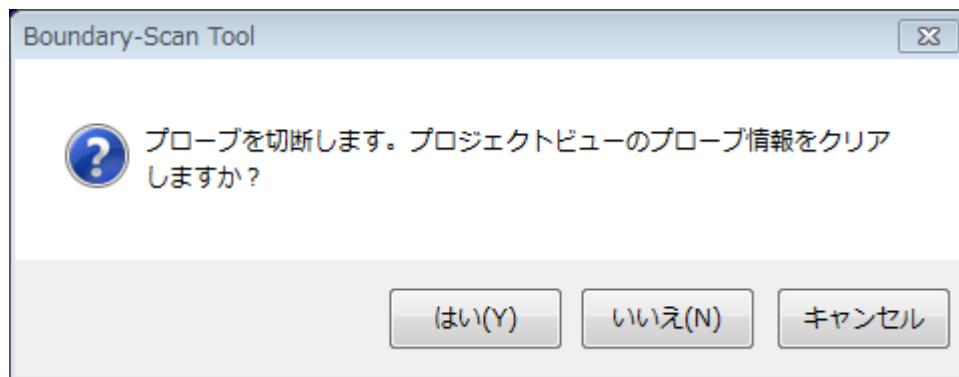
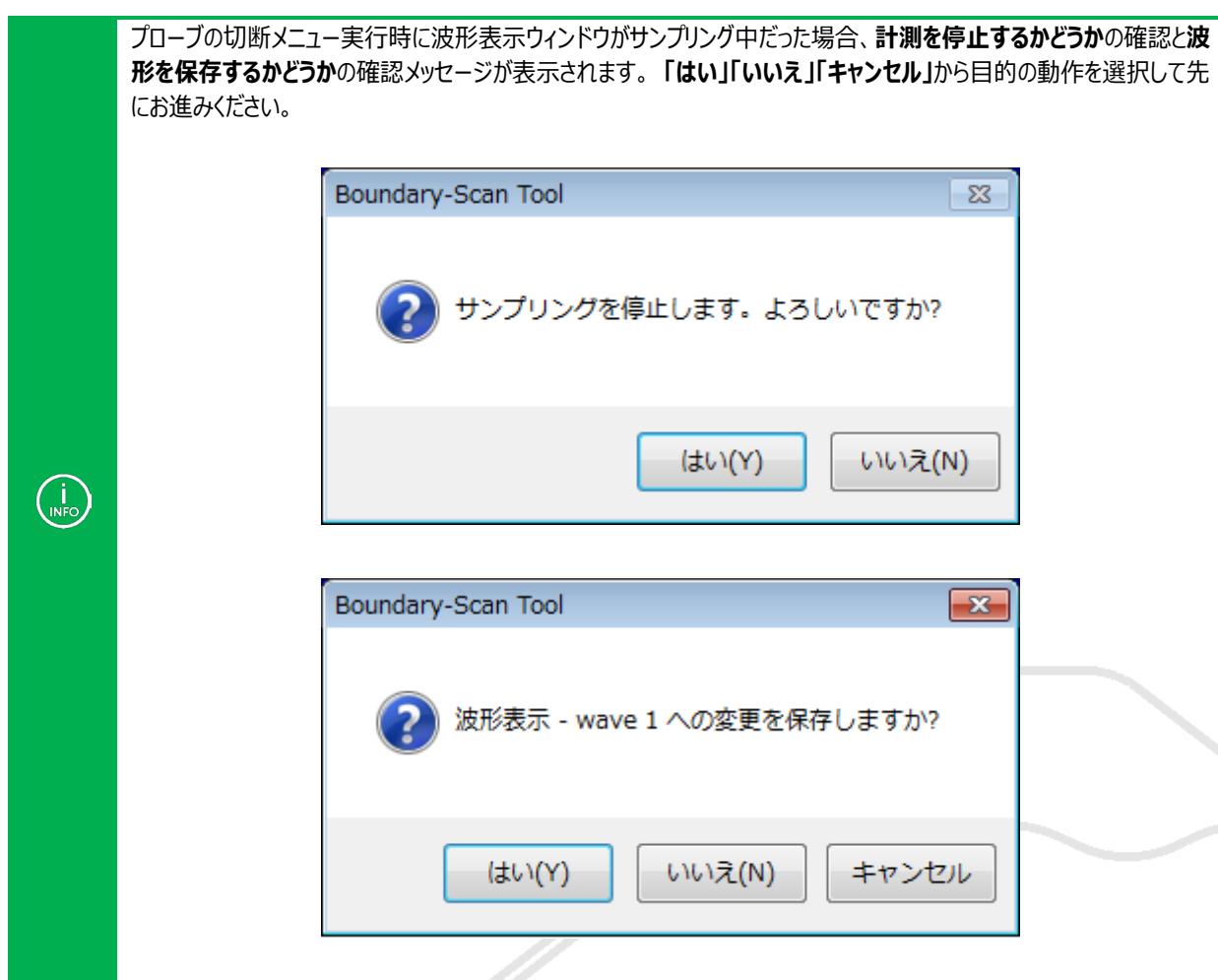


Figure 13

プローブの切断メニュー実行時に波形表示ウインドウがサンプリング中だった場合、計測を停止するかどうかの確認と波形を保存するかどうかの確認メッセージが表示されます。「はい」「いいえ」「キャンセル」から目的の動作を選択して先にお進みください。



2.2.15. JTAG クロックの設定

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → JTAG クロックの設定	---
---	*プロジェクトビュー → プローブ 右クリックによるコンテキストメニューでも操作可能	---

バウンダリスキャンテストを実行する際の、プローブとターゲットデバイス間のクロック周波数を設定します。上記メニューを選択すると"JTAG クロックの設定"ダイアログが表示されますので、ターゲットに合わせてクロック周波数を設定してください。

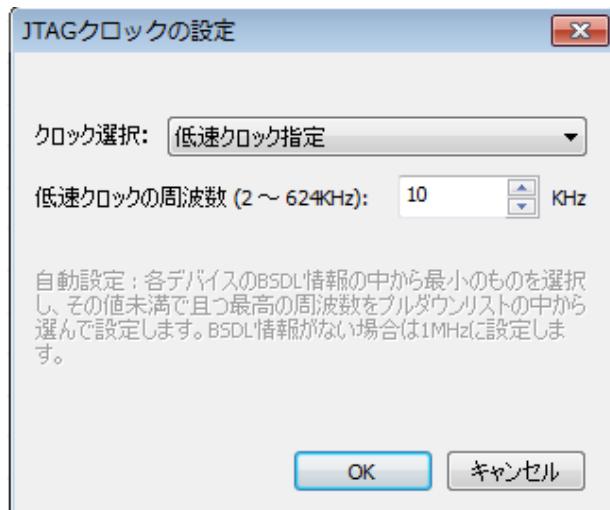


Figure 14



本ソフトウェアは、デバイスの自動認識、整合性テストを"低速クロック"の"10kHz"で行います。その後、**自動設定**により JTAG クロックを上げます。

2.2.16. JTAG クロックの選択

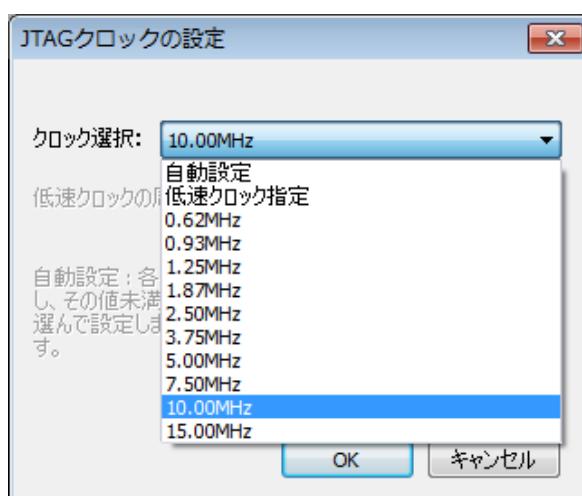


Figure 15

プローブに設定できる周波数がプルダウントリストに表示されます。

プロジェクトの初期値は"低速クロック指定"になります。

各項目の詳細は以下の通りです。

Table 9

自動設定	<p>クロック周波数を自動で設定します。 自動認識できるターゲットに接続されているかプロジェクトにデバイスが設定されていれば、各デバイスの BSDL 情報の中から最低の周波数を選び、その値未満で且つプルダウントリストから選択できる最高の周波数を設定します。 BSDL 情報がひとつもなければ"0.62MHz"を設定します。</p> <div style="border: 1px solid green; padding: 5px;"> i INFO 例えば A と B のデバイスがあり、BSDL にそれぞれ 12MHz と 20MHz が記述されていたとする、上図のプルダウントリストでは（12MHz 未満で最大のクロック周波数）10MHz が選択されます。 </div>
低速クロック指定	<p>625kHz 未満の周波数をプルダウントリスト下の低速クロックエディットボックスで設定します。</p> <div style="border: 1px solid green; padding: 5px;"> i INFO プロジェクトの初期値として、"低速クロック指定"の"10kHz"が設定されています。 </div>
他の周波数	<p>選択された周波数で内部的にターゲットとの通信テストを行います。動作すればその周波数に設定します。動作しなければ動作する周波数まで値を下げます。</p> <div style="border: 1px solid green; padding: 5px;"> i INFO BSDL 情報とは無関係に動作するかしないかで値を設定するため、BSDL に記述されている動作周波数より大きい周波数に設定されることがあります。 </div>

2.2.17. JTAG クロックの設定（低速）

低い周波数でプローブとターゲットデバイスとの JTAG 通信を行います。クロック選択プルダウントリストで"低速クロック指定"を選択すると有効になります。値の範囲は 2~624（単位：kHz）です。

2.2.18. 整合性テスト

ツールボタン	メニューバーの操作	ショートカットキー
	プロジェクト → 整合性テスト *プロジェクトビュー → ボード → 任意のデバイス 右クリックによる コンテキストメニューでも操作可能	---

プロジェクトビューに構成されたバウンダリスキャンチェーンでバウンダリスキャンテストを行うことができるかをテストします。テストに成功すると上記メニューはグレーアウト状態になり、以下の機能が使用可能になります。

- ・ 波形表示ウィンドウ
- ・ 端子状態一覧ウィンドウ
- ・ 接続テストウィンドウ

『デバイスの自動認識』に成功したときは整合性テストに通過したと見なされ、上記メニューはグレーアウト状態になります。
整合性テストに通過している状態で、"デバイスの追加"・"デバイスの編集"・"デバイスの削除"・"デバイスの上下移動"メニューを実行しようとすると、以下の警告メッセージが表示されます。プロジェクト側のデバイス構成が変わることで整合性テストに失敗する可能性があります。

 Boundary-Scan Tool

 整合性テストを通過済みですが処理を続けますか？

OK キャンセル

2.2.19. ポートグループの追加

プロジェクトビューの「ポートグループ」ツリーにグループを作成することができます。作成したグループは波形表示ウインドウや端子状態一覧ウインドウなどで使うことができます。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → ポートグループを追加	
	*プロジェクトビュー → ポートグループ 右クリックによるコンテキストメニューでも操作可能	Ctrl + G

ボード上の全ポートから組み合わせを指定して、グループを作成できます。
グループを作成すると波形表示ウインドウや端子状態一覧ウインドウで使用することができます。

- **グループ作成ダイアログ**



Figure 16

- **名前を指定する**

「名前」グループ名を指定します

- **構成するポートを指定する**

「すべてのポート」リストにボード上のすべてのポートがリストされます。グループに追加したいポートをそのリストで選択し[←]ボタンをクリックしてグループに追加します。「構成」リストの最後に追加されます。

- **構成するポートの順序を変える**

「構成」リストで順番を変更したいポートを選択し[↑]ボタンまたは[↓]ボタンをクリックしてポートの順番を変更します。

- **構成するポートを削除する**

「構成」リストで削除したいポートを選択し[×]ボタンをクリックしてポートを削除します。

- **値を表示するフォーマットを指定する**

「16進数」または「2進数」をクリックすることでグループの値を表示するフォーマットを指定できます。波形表示ウインドウや端子状態一覧ウインドウで表示するときのデフォルトの表示フォーマットになります。

- **グループを追加する**

[OK]ボタンをクリックしてグループを追加します。最後に追加されます。

2.2.20. ポートグループの編集

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ポートグループ → 追加された ポートグループ 右クリックによる コンテキストメニュー → グループの編集 で操作可能	---

「グループを追加する」で追加されたグループの名前・ポートの構成・表示フォーマットを変更することができます。
上記メニューを選択するとグループ作成メニューが開きますので、「グループを追加する」を同様の手順でグループを編集します。

2.2.21. ポートグループの削除

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ポートグループ → 追加された ポートグループ 右クリックによる コンテキストメニュー → 削除 で操作可能	---

プロジェクトビューから選択したグループを削除することができます。

2.2.22. デバイス参照名の設定

デバイスにデバイス参照名を設定できます。設定された名前を接続テストウィンドウで使うことができます。

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。	---
---	*プロジェクトビュー → ボード → 追加された デバイス 右クリックによる コンテキストメニュー → デバイス参照名の設定 で操作可能	---

デバイス参照名の設定ダイアログが開きます。デバイス参照名を入力し、[OK]ボタンをクリックします。

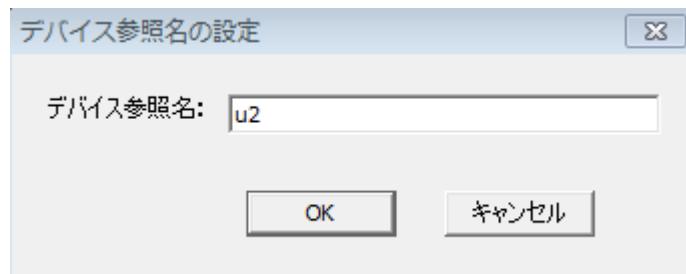


Figure 17

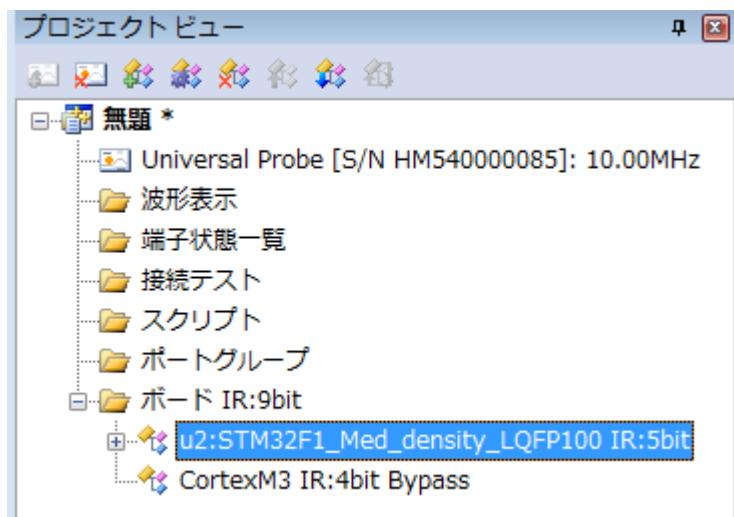


Figure 18

また、下記の方法でも変更できます。

プロジェクトビュー → ボード → 追加された デバイス 右クリックによるコンテキストメニュー → デバイスプロパティ

デバイスプロパティが開きます。デバイス参照名を編集し[OK]ボタンをクリックします。

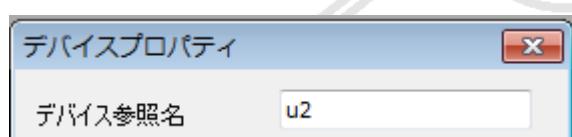


Figure 19

2.2.23. デバイス参照名のリセット

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス 右クリックによる コンテキストメニュー → デバイス参照名のリセット で操作可能	---

設定されたデバイス参照名を解除します。

2.2.24. ポート名の変更（デバイス単位）

ポートの識別を容易にするためにデバイス単位、ポート単位にポート名を変更できます。また、変更したポート名を保存できます。

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス 右クリックによる コンテキストメニュー → ポート名の変更 で操作可能	---

ポート名の変更ダイアログが開きます。定義ファイル欄にポート名を定義したファイル名を入力し、[OK]ボタンをクリックします。

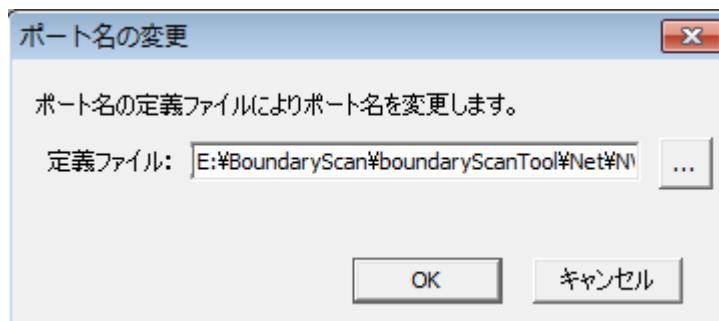


Figure 20

ポート名の定義ファイルのフォーマットは以下の通りです。

Table 10

[PortName]	ポートごとのピン名、ポート名のリストです。 "ピン名,ポート名" の組み合わせです。 例： [PortName] 23 , INT0 30 , INT2 32 , INT4 35 , INT3
-------------------	---



ポート名の定義ファイルにエラーがあると、変更しようとしたデバイスのポート名は全てリセットされますので注意してください。

2.2.25. ポート名の変更（ポート単位）

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス → ポート 右クリックによる コンテキストメニュー → ポート名の変更 で操作可能	---

ポート名の変更ダイアログが開きます。ポート名欄にポート名を入力し、[OK]ボタンをクリックします。

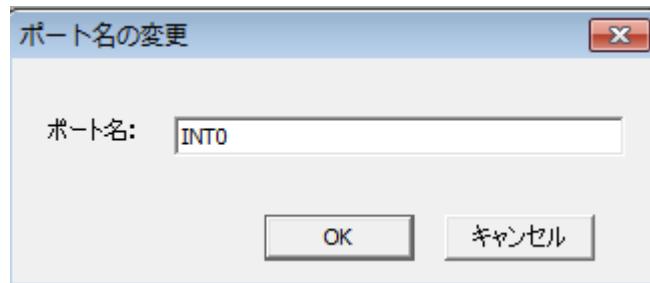


Figure 21

2.2.26. ポート名のリセット（デバイス単位）

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス 右クリックによる コンテキストメニュー → ポート名のリセット で操作可能	---

変更したポート名がデバイス単位で解除されます。ポート名は BSDL ファイルで定義された名前になります。

2.2.27. ポート名のリセット（ポート単位）

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス → ポート 右クリックによる コンテキストメニュー → ポート名のリセット で操作可能	---

変更したポート名が解除されます。ポート名は BSDL ファイルで定義された名前になります。

2.2.28. ポート名変更の保存

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された デバイス 右クリックによる コンテキストメニュー → ポート名変更の保存 で操作可能	---

ポート名変更保存ダイアログが開きます。定義ファイル欄にファイル名を入力し、[OK]ボタンをクリックします。

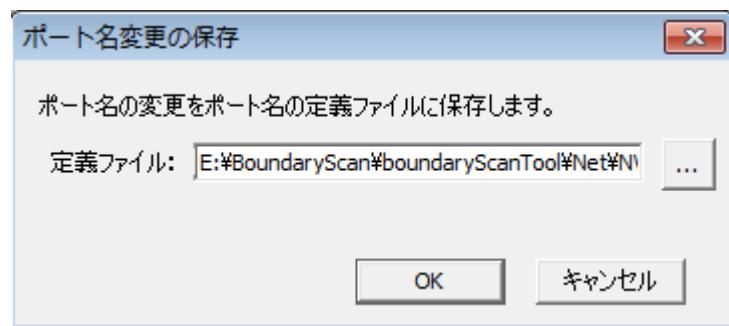


Figure 22

2.3. 波形表示ウィンドウ

2.3.1. 波形ウィンドウの新規作成

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 波形ウィンドウの新規作成	
---	* プロジェクトビュー → 波形表示 → 右クリックによる コンテキストメニュー → 波形表示 でも操作可能	Ctrl + W

上記メニューを実行すると、プロジェクトビューに "wave n" が追加されるとともに波形表示ウィンドウが開きます。（n は 1 からの通し番号）上記メニューを実行するたびに、新たに空の "wave n" が追加され、追加された空の波形がウィンドウに表示されます。ただし、プロジェクトビュー上で "wave n" を選択した状態で上記メニューを実行すると、新たな波形は作成せず、選択した波形をウィンドウに表示します。現在開いているリストは波形表示ウィンドウのタイトルバーに表示されます。

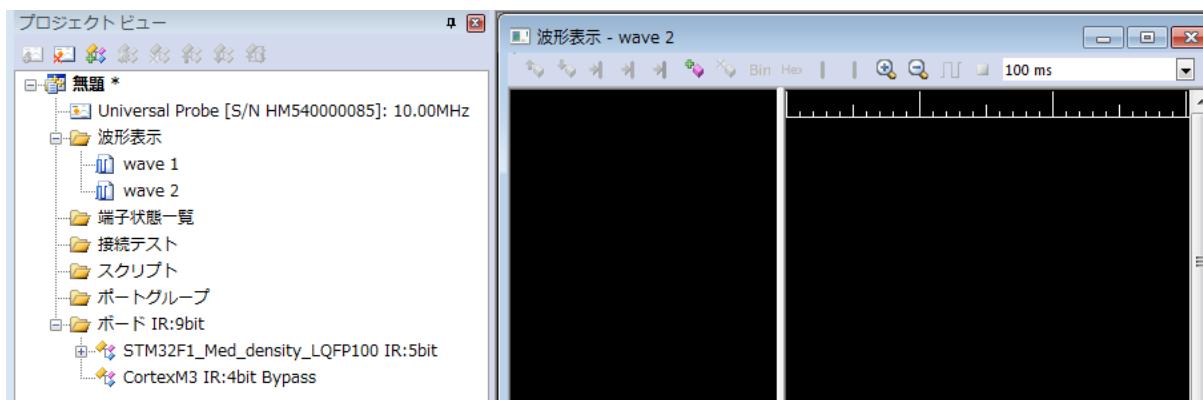


Figure 23

2.3.2. 波形をビューワで開く

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 波形をビューワで開く	---

上記メニューを実行すると、ファイル選択ダイアログが表示され、保存してある波形ファイル（拡張子は.bsw）を選択すると波形ウインドウでその波形を開きます。波形を開いた場合、以下の機能は使用できなくなります。

- ポートの追加
- グループの作成
- ポートグループの削除
- サンプリング開始
- サンプリング停止
- サンプリング間隔指定

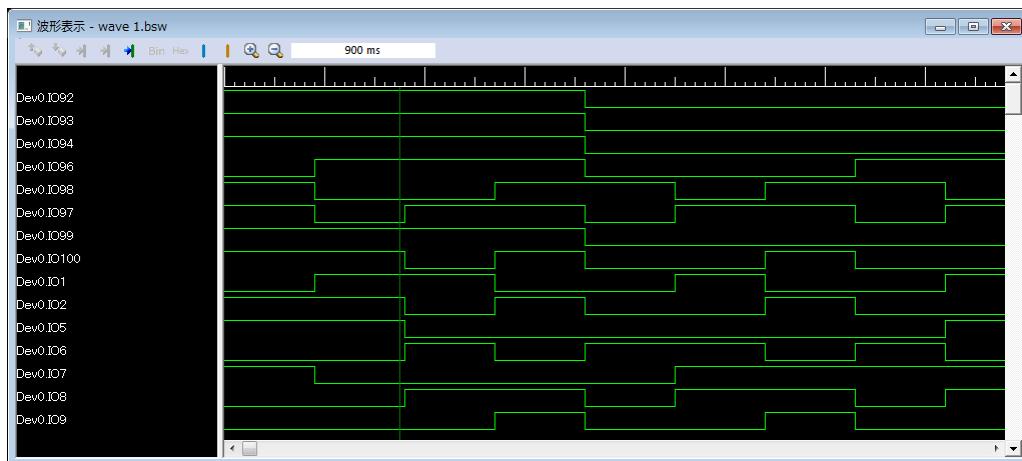


Figure 24

2.3.3. 波形の名前変更

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。	
---	*プロジェクトビュー → ボード → 追加された [wave n] 右クリックによる コンテキストメニュー → 名前変更 で操作可能	
---	*[wave n] を選択した状態で クリック 又は F2 キー で操作可能	

上記メニューを選択すると、プロジェクトビューの該当項目が編集状態になりますので、目的に合わせて名称を変更してください。

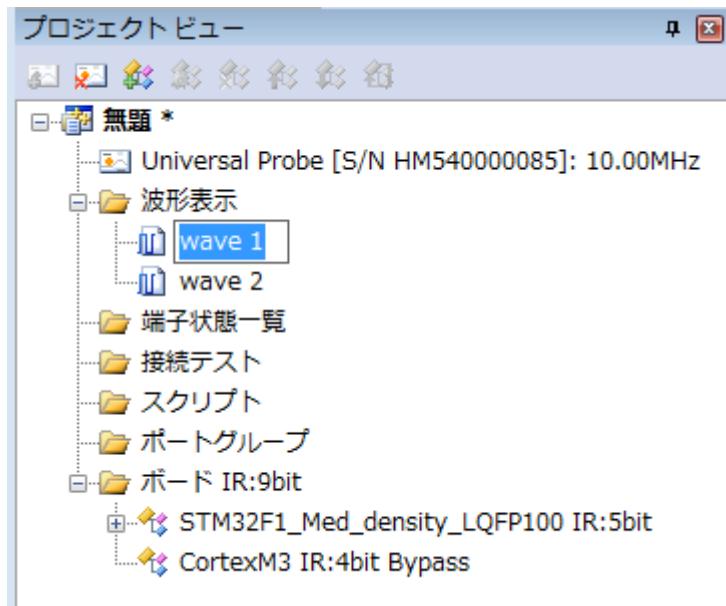


Figure 25

2.3.4. 波形の削除

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → ボード → 追加された [wave n] 右クリックによる コンテキストメニュー → 削除 で操作可能	---

上記メニューを選択すると、プロジェクトビューの該当項目が削除されます。

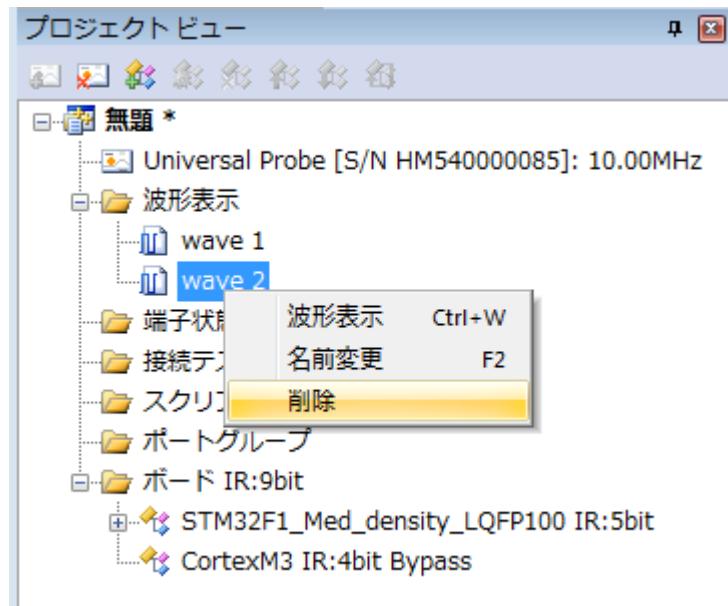
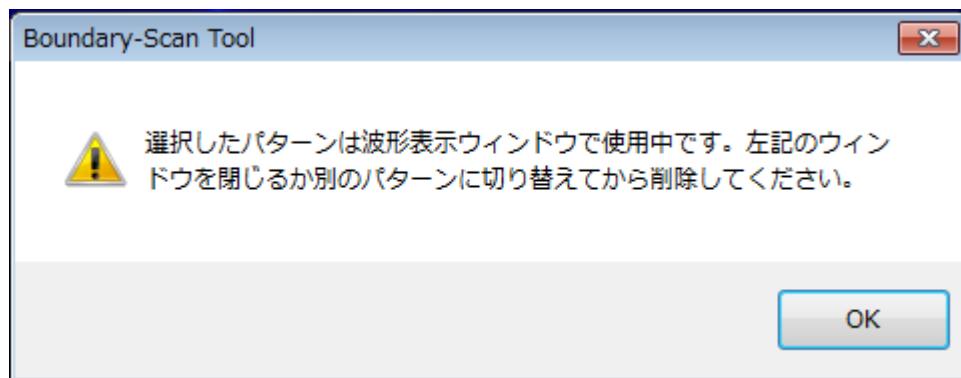


Figure 26

上記メニューを選択したときに、該当するリストが波形表示ウィンドウに表示されているリストだった場合は、下記のエラーメッセージが表示されます。



2.3.5. ポート（信号）の追加

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポート/ポートグループ名表示領域 → 右クリックによる コンテキストメニュー → 追加 で操作可能	---

上記メニューを選択すると"ポートの追加"ダイアログが表示されます。ダイアログにはプロジェクトに登録されている全デバイスの全ポートが表示されていますので、状態を見たいポートにチェックを付けて"OK"ボタンをクリックしてください。ポートは複数選択が可能です。

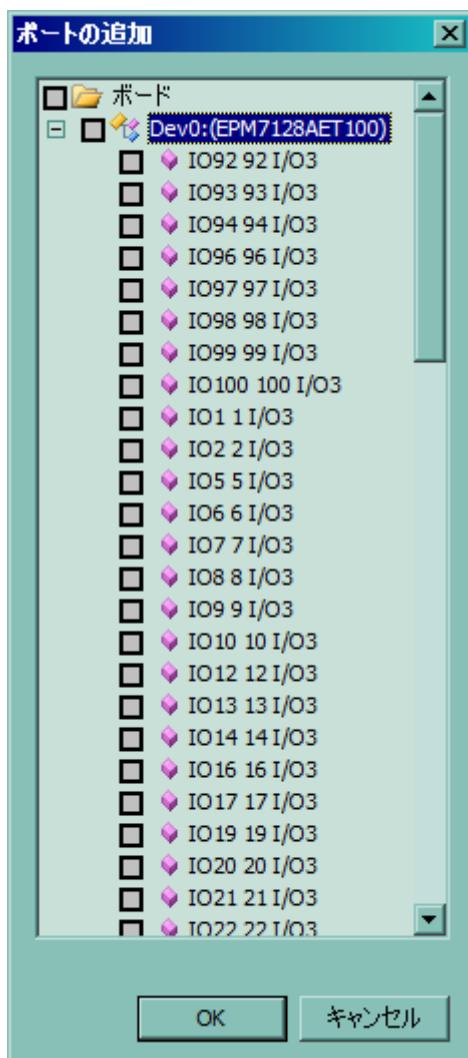


Figure 27

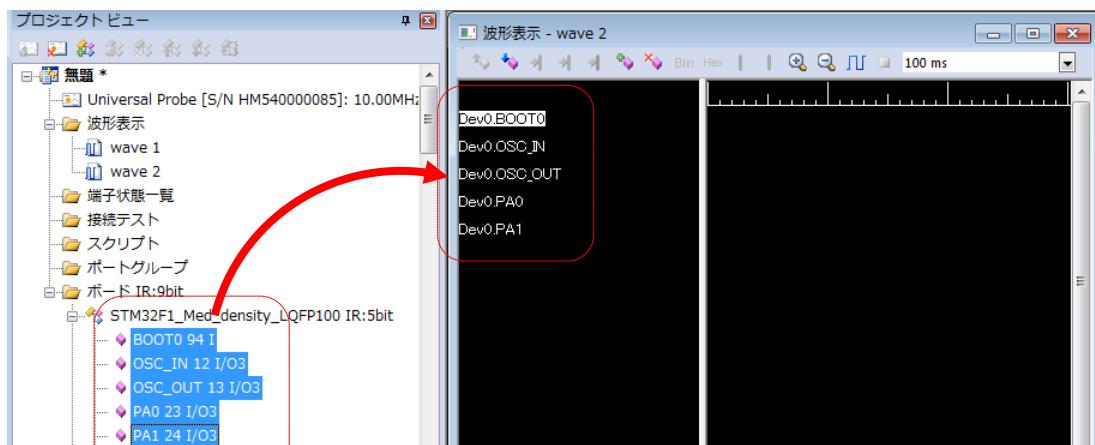
"ポート"タグのチェックボックスをクリックすると、全デバイスの全ポートのチェックが切り替わります。また、"Dev n"(n はバウンダリスキヤンチェーンの通し番号)タグのチェックボックスをクリックすると、そのデバイスの全ポートのチェックが切り替わります。

ポートの中にはコントロールセルを共有しているものがあります。

(BSDL の"function"が"control"となっている番号のセルを、複数のポートで共有し、入出力方向・出力の許可/禁止を一括管理している)

そのようなポートのひとつにチェックを付けた場合、共有している他のポートにもチェックが付きます。

ポートの追加はプロジェクトビューから端子状態一覧ウィンドウへのドラッグ&ドロップでも行うことができます。プロジェクトビュー上では"CTRL+左クリック"・"SHIFT+左クリック"で複数選択が可能です。こちらもコントロールセルを共有しているポートを追加した場合は共有している他のポートも併せて追加されます。



2.3.6. ポートグループ（信号）の作成

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *波形表示ウィンドウ → ポート/ポートグループ名表示領域 → 右クリックによる コンテキストメニュー → グループ作成 で操作可能	---

追加したポートの組み合わせを指定して、グループを作成できます。
グループを作成すると、2進数/16進数で値の指定、表示ができます。
このメニューで作成したグループはプロジェクトビューには追加されません。

グループ作成ダイアログでのポートグループ作成方法は [2.2.19. ポートグループの追加](#)を参照してください。

ポートグループ名表示領域は、ポートに対して 下図のようにデバイス参照名とポート名を表示しています。グループはグループ名のみ表示します。



Figure 28

2.3.7. ポートグループ（信号）の2進/16進表示 切り替え

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポートグループ選択 → 右クリックによる コンテキストメニュー → “2進数で表示”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポートグループ選択 → 右クリックによる コンテキストメニュー → “16進数で表示”で操作可能	---

選択したグループの値を2進数または16進数で表示するか切り替えます。

2.3.8. ポート/ポートグループ（信号）の表示順序変更

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “上へ”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “下へ”で操作可能	---

ポートグループ名表示領域で移動させたいポートまたはグループを1つ選択し移動させることができます。

2.3.9. ポート/ポートグループ（信号）の削除

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “削除”で操作可能	---

ポートグループ名表示領域で削除したいポートまたはグループを1つ選択し削除することができます。

2.3.10. サンプリング周期の選択

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *波形表示ウィンドウ → ツールバー → サンプリング周期指定コンボボックス 選択	---

波形表示ウィンドウでは選択したサンプリング周期ごとにサンプリングを行います。

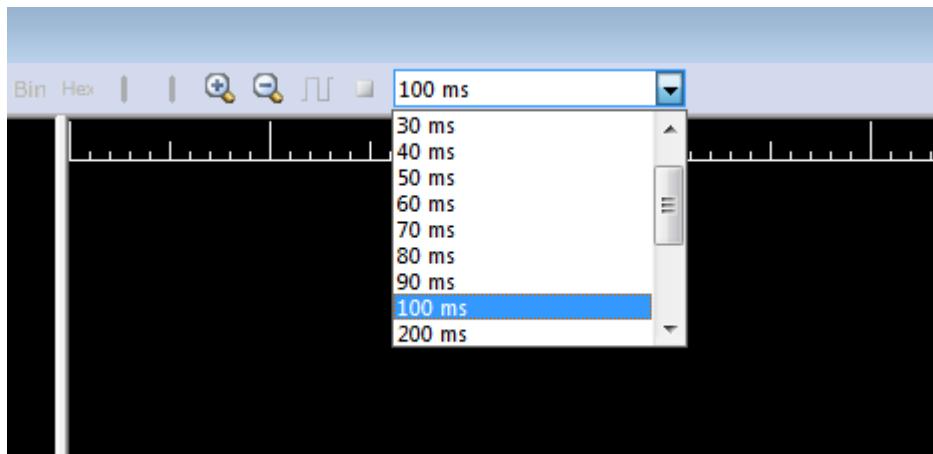


Figure 29

選択できるサンプリング周期は以下のとおりです。

- 10ms
- 20ms
- 30ms
- 40ms
- 50ms
- 60ms
- 70ms
- 80ms
- 90ms
- 100ms
- 200ms
- 300ms
- 400ms
- 500ms
- 600ms
- 700ms
- 800ms
- 900ms



JTAG クロックのスピードにより、選択できるサンプリング周期が変化します。

2.3.11. サンプリング開始（計測開始）

サンプリング（計測）を開始します。

ツールボタン	メニューバーの操作	ショートカットキー
	波形表示 → サンプリング開始	F7

2.3.12. サンプリング停止（計測停止）

サンプリング（計測）を停止します。

ツールボタン	メニューバーの操作	ショートカットキー
	波形表示 → サンプリング停止	Ctrl + F7

2.3.13. ズームイン

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → 波形表示領域 → 右クリックによる コンテキストメニュー で操作可能	---

波形表示領域の波形表示を拡大します。

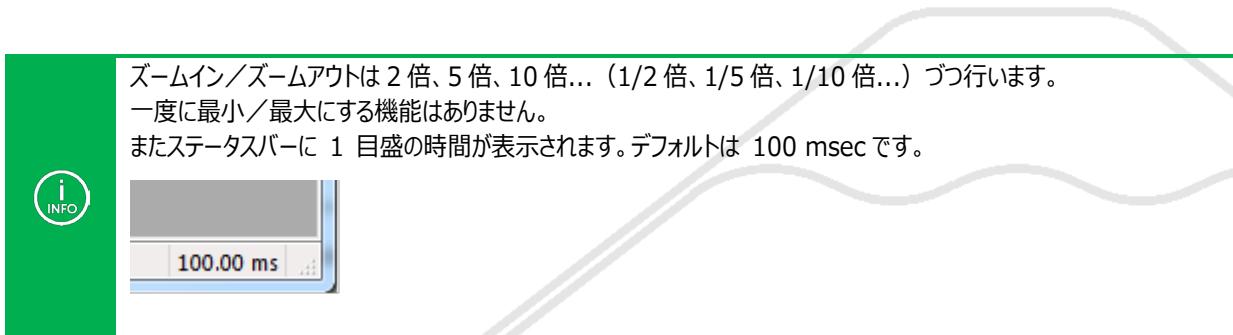
波形表示領域にカーソルがある場合はカーソル、ない場合は現在の表示範囲の中心を軸に拡大します。

2.3.14. ズームアウト

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *波形表示ウインドウ → 波形表示領域 → 右クリックによる コンテキストメニュー で操作可能	---

波形表示領域の波形表示を縮小します。

波形表示領域にカーソルがある場合はカーソル、ない場合は現在の表示範囲の中心を軸に縮小します。



2.3.15. カーソルの設定

波形表示ウインドウ→波形表示領域内で左クリックすることでカーソルを設定できます。
 緑の実線がカーソルを示します。

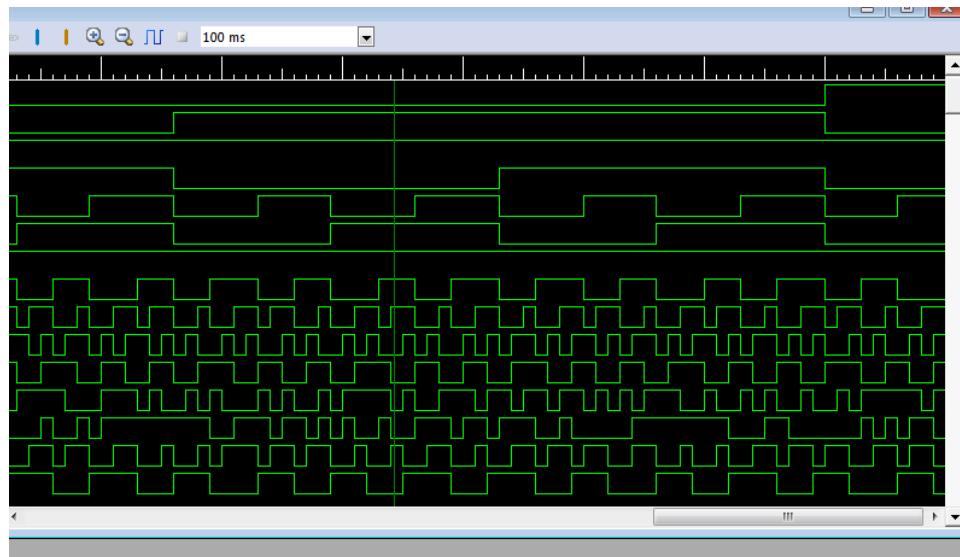


Figure 30

また、ステータスバーにカーソル位置(C)を表示します。

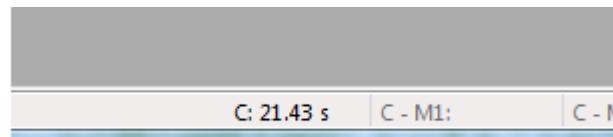
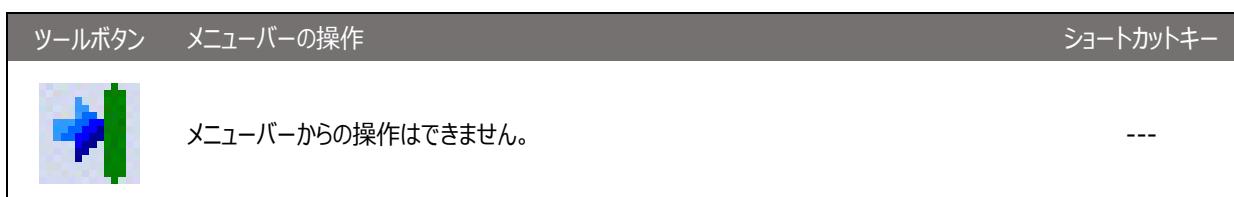


Figure 31

2.3.16. カーソルの移動

左クリック 又は カーソルをドラッグすることにより移動させることができます。

2.3.17. カーソル位置を中央に表示

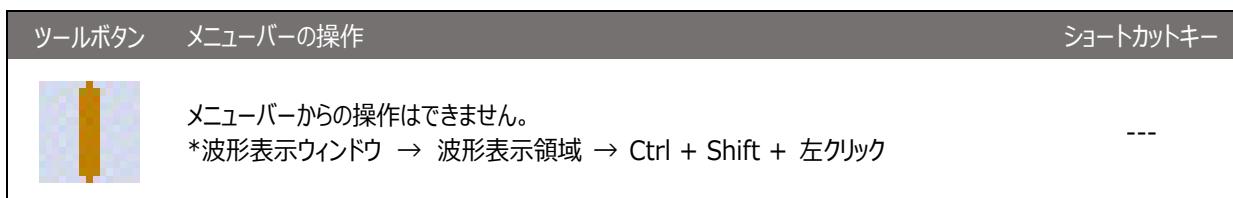
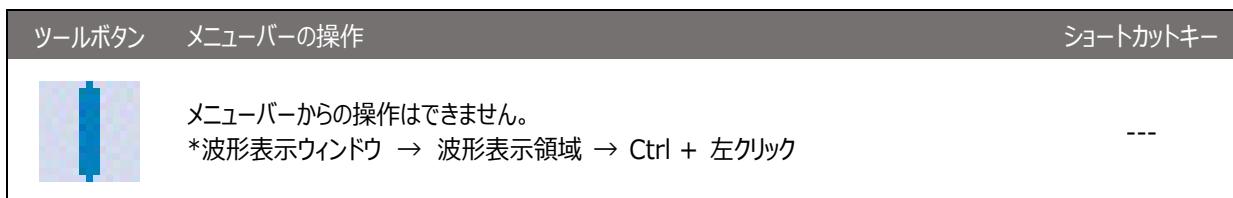


カーソルの位置が波形表示範囲の中央にくるように表示されます。

i INFO	マーカと一緒に使うことによりカーソルとマーカの間の時間を検証したりすることができるようになります。
--	---

2.3.18. マーカの設定

波形表示領域内を 2箇所マークすることができます。（マーク 1、マーク 2）



波形表示領域の波形表示を縮小します。

波形表示領域にカーソルがある場合はカーソル、ない場合は現在の表示範囲の中心を軸に縮小します。

ツールバーのボタンによりカーソル位置にマーカを設定することができます。また、Ctrl キーもしくは Ctrl キーと Shift キーを押しながら左クリックすることにより任意の波形範囲内の位置にマーカを設定することができます。

マーク 1 は青、マーク 2 は橙の破線で表示します。

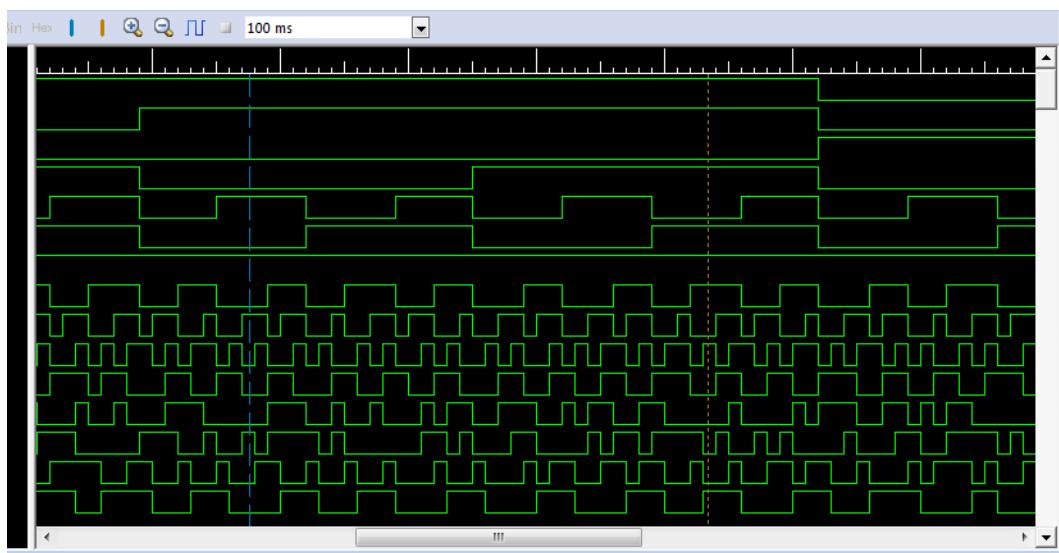


Figure 32

ステータスバーにそれぞれのマーカの位置(M1、M2)を表示します。また、カーソル間、各マーカ間の差分もステータスバーに表示します。

C - M1 :カーソル - マーカ 1 間の差分です。

C - M2 :カーソル - マーカ 2 間の差分です。

M1 - M2 :マーカ 1 - マーカ 2 間の差分です。



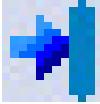
Figure 33

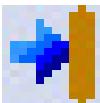
2.3.19. マーカ位置へカーソルを移動

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。	---
---	*波形表示ウインドウ → 波形表示領域 → 右クリックによる コンテキストメニュー → 「カーソルをマーカ 1 に移動」を選択	---

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。	---
---	*波形表示ウインドウ → 波形表示領域 → 右クリックによる コンテキストメニュー → 「カーソルをマーカ 2 に移動」を選択	---

2.3.20. マーカ位置を中央に表示

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	---

マーカの位置が波形表示範囲の中央にくるように表示されます。

2.3.21. 波形の印刷

波形表示ウィンドウの現在の表示範囲を印刷することができます。

詳細は「[3.1.14. 印刷レビュー](#)」, 「[3.1.15. プリンターの設定](#)」, 「[3.1.16. 印刷](#)」を参照してください。

2.3.22. 波形の保存

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 波形の保存	---

計測した波形データを、ファイルに保存する行うことができます。

ファイルの拡張子は .bsw となります。



保存したファイルはメニューの **ファイル → 波形をビューアで開く** で読み込むことができます。

2.4. 端子状態一覧ウィンドウ

2.4.1. 端子状態一覧の新規作成/開く

空の端子状態一覧リストを作成して端子状態一覧ウィンドウに表示します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 端子状態一覧の新規作成/開く *プロジェクトビュー → 端子状態一覧 右クリックによる コンテキストメニューでも操作可能	Ctrl + L

上記メニューを実行すると、プロジェクトビューに"list n"が追加されるとともに端子状態一覧ウィンドウが開きます。
(n は 1 からの通し番号) 上記メニューを実行するたびに、新たに空の"list n" が追加され、追加された空のリストがウィンドウに表示されます。ただし、プロジェクトビュー上で"list n" を選択した状態で上記メニューを実行すると、新たにリストは作成せず、選択したリストをウィンドウに表示します。現在開いているリストは端子状態一覧ウィンドウのタイトルバーに表示されます。

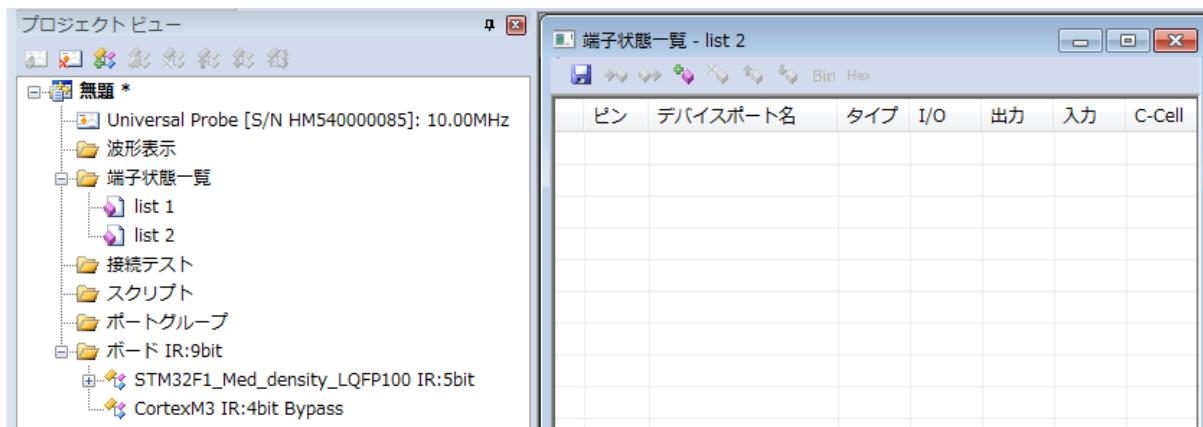


Figure 34

2.4.2. リストの名前変更

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → 端子状態一覧 → 追加された [list n] 右クリックによる コンテキストメニュー → 名前変更 で操作可能 *[list n] を選択した状態で クリック 又は F2 キー で操作可能	---

上記メニューを選択すると、プロジェクトビューの該当項目が編集状態になりますので、目的に合わせて名称を変更してください。

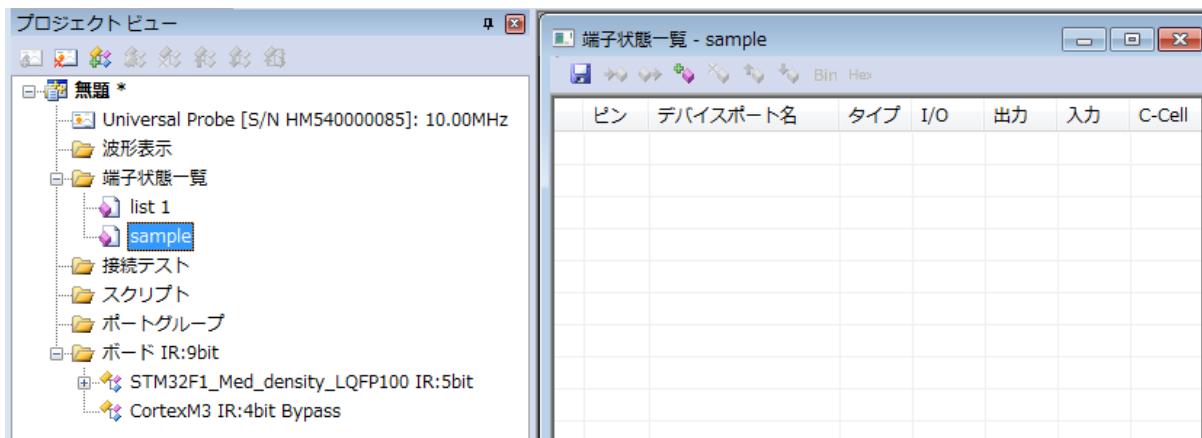
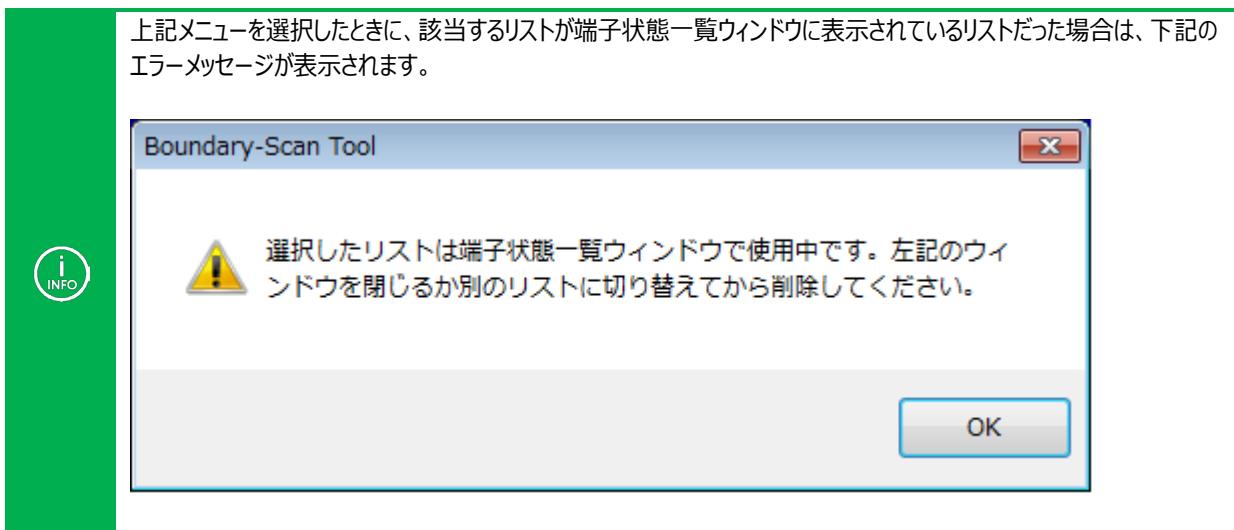


Figure 35

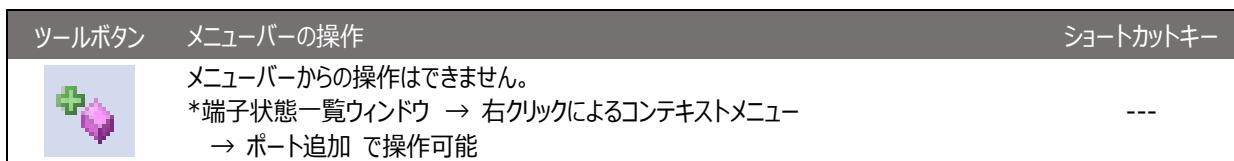
2.4.3. リストの削除

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	
---	*プロジェクトビュー → 端子状態一覧 → 追加された [list n] 右クリックによる コンテキストメニュー → 削除 で操作可能	---

上記メニューを選択すると、プロジェクトビューの該当項目が削除されます。



2.4.4. ポートの追加



上記メニューを選択すると"ポートの追加"ダイアログが表示されます。ダイアログにはプロジェクトに登録されている全デバイスの全ポート及びグループが表示されていますので、状態を見たいポートやグループにチェックを付けて"OK"ボタンをクリックしてください。ポートは複数選択が可能です。

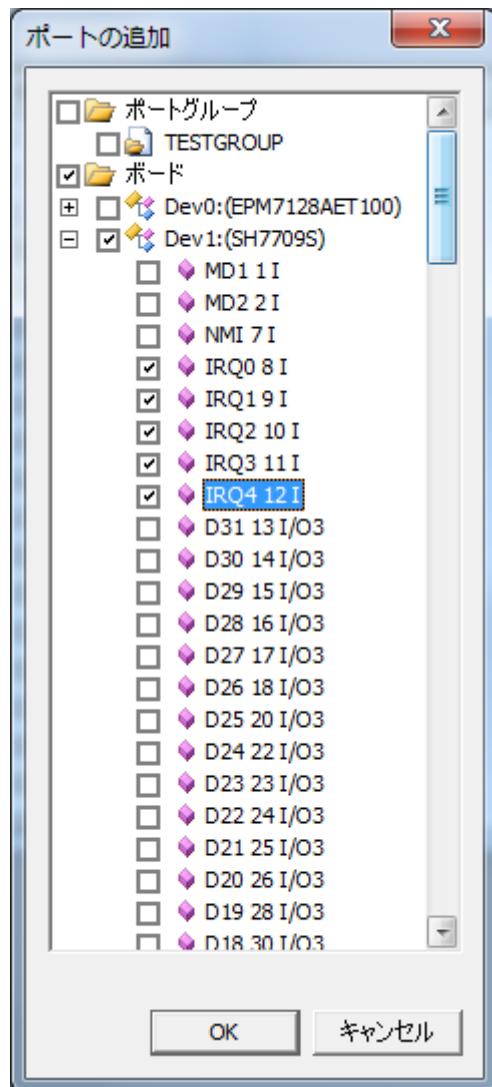


Figure 36

"ポート"タグのチェックボックスをクリックすると、全デバイスの全ポートのチェックが切り替わります。また、**"Dev n"**(n はバウンダリスキヤンチェーンの通し番号)タグのチェックボックスをクリックすると、そのデバイスの全ポートのチェックが切り替わります。

ポートの中にはコントロールセルを共有しているものがあります。

(BSDL の"function"が"control"となっている番号のセルを、複数のポートで共有し、入出力方向・出力の許可/禁止を一括管理している)

そのようなポートのひとつにチェックを付けた場合、共有している他のポートにもチェックが付きます。

ポートグループタグのチェックボックスをクリックすると、全グループのチェックが切り替わります。

OK ボタンによりグループが追加されるとき、**グループを構成しているポートと そのポートとコントロールセルを共有して**

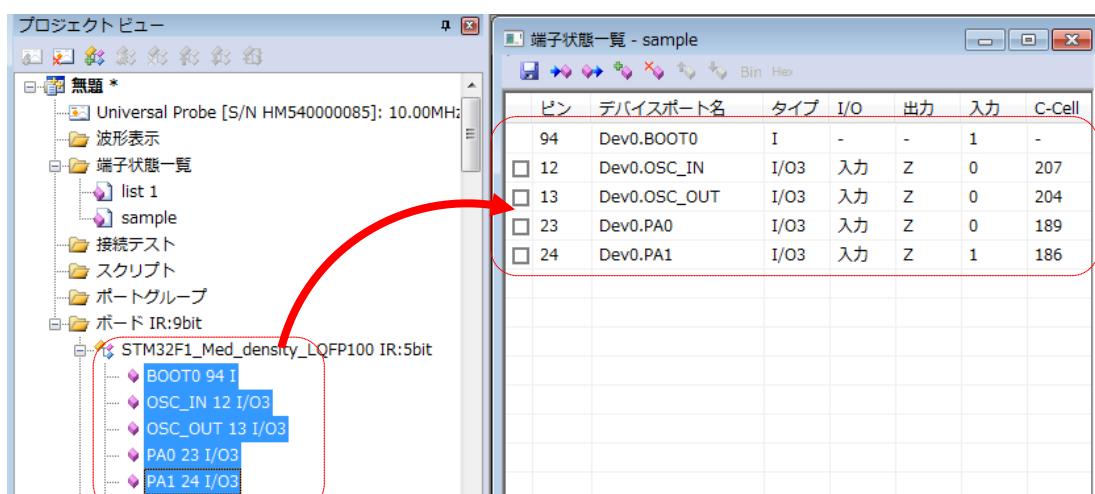
いるポートが併せて追加されます。

ポートやグループの追加はプロジェクトビューから端子状態一覧ウィンドウへのドラッグ&ドロップでも行うことができます。

プロジェクトビュー上では"CTRL+左クリック"・"SHIFT+左クリック"で複数選択が可能です。

こちらもコントロールセルを共有しているポートを追加した場合は共有している他のポートも併せて追加されます。

グループをドラッグ&ドロップした場合はグループを構成しているポートとそのポートとコントロールセルを共有しているポートが追加されます。



2.4.5. ポートグループの作成

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → 右クリックによるコンテキストメニュー → グループ作成 で操作可能	---

追加したポートの組み合わせを指定して、グループを作成できます。

グループを作成すると、2進数/16進数で値の指定、表示ができます。

このメニューで作成したグループはプロジェクトビューには追加されません。

グループ作成ダイアログでのポートグループ作成方法は [2.2.19. ポートグループの追加](#)を参照してください。

2.4.6. ポートグループの 2 進/16 進表示切り換え

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → ポートグループ選択 → 右クリックによる コンテキストメニュー → “2 進数で表示”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → ポートグループ選択 → 右クリックによる コンテキストメニュー → “16 進数で表示”で操作可能	---

選択したグループの値を 2 進数または 16 進数で表示するか切り替えます。

2.4.7. ポート/ポートグループの表示順序変更

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “上へ”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “下へ”で操作可能	---

ポートグループ名表示領域で移動させたいポートまたはグループを 1 つ選択し移動させることができます。

2.4.8. ポート/ポートグループの削除

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *端子状態一覧ウィンドウ → ポート/ポートグループ選択 → 右クリックによる コンテキストメニュー → “ポート削除”で操作可能	---

上記メニューを選択すると“ポートの削除”ダイアログが表示されます。ダイアログには端子状態一覧ウィンドウに登録されている全ポートとグループが表示されていますので、削除したいポート及びグループにチェックを付けて“OK”ボタンをクリックしてください。ポートは複数選択が可能です。

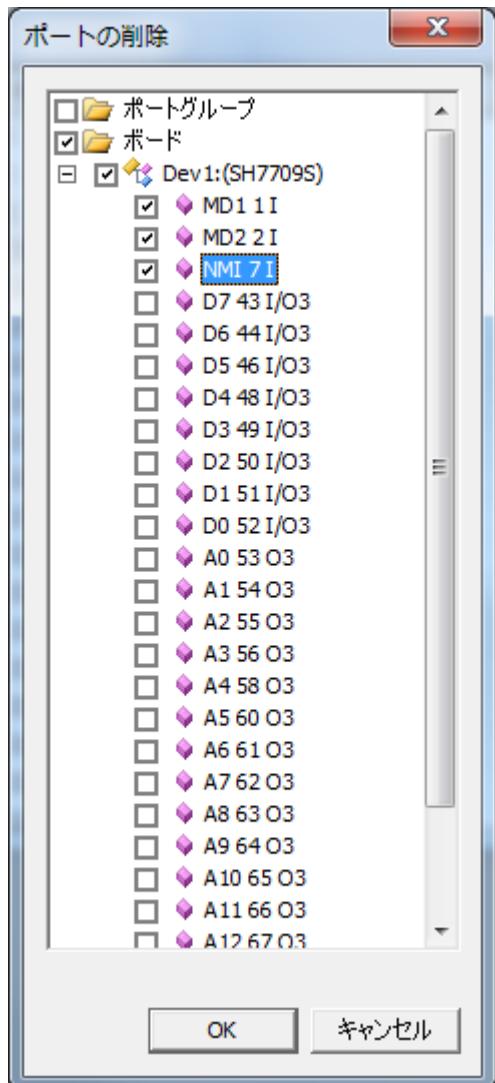
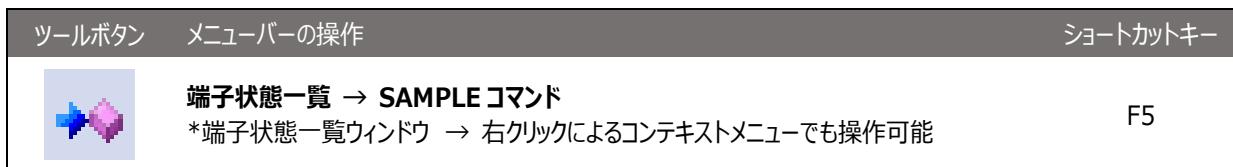


Figure 37



"ポート追加"と同様、"ボード"タグのチェックボックスをクリックすると、全デバイスの全ポートのチェックが切り替わります。また、"Dev n"(n はバウンダリスキャンチェーンの通し番号)タグのチェックボックスをクリックすると、そのデバイスの全ポートのチェックが切り替わります。コントロールセルを共有しているポートも、"ポート追加"と同様にして一斉にチェックが切り替わります。削除したポートがグループの構成要素だった場合、そのグループはチェックをつけていなくても削除されます。逆にグループのみチェックをつけた場合は、構成要素のポートは削除されません。

2.4.9. SAMPLE コマンド実行

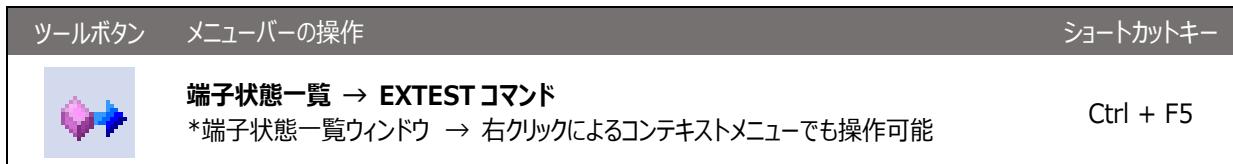


バウンダリスキャンチェーン上の各デバイスに SAMPLE 命令を与えて、デバイスの入出力データをバウンダリスキャンレジストに取り込みます。取り込ませた値を本ソフトウェアが取り出し、端子状態一覧ウィンドウのリストを更新します。更新の前後で値が変わった行は、表示色が変更されます。

ピン	デバイスポート名	タイプ	I/O	出力	入力	C-Cell
1	Dev0.MD1	I	-	-	0	-
2	Dev0.MD2	I	-	-	0	-
7	Dev0.NMI	I	-	-	1	-
8	Dev0.IRQ0	I	-	-	0	-
9	Dev0.IRQ1	I	-	-	0	-
10	Dev0.IRQ2	I	-	-	0	-
11	Dev0.IRQ3	I	-	-	0	-
12	Dev0.IRQ4	I	-	-	0	-
□ 13	Dev0.D31	I/O3	入力	Z	1	266
□ 14	Dev0.D30	I/O3	入力	Z	1	265
□ 15	Dev0.D29	I/O3	入力	Z	1	264
□ 16	Dev0.D28	I/O3	入力	Z	1	263
□ 17	Dev0.D27	I/O3	入力	Z	1	262
□ 18	Dev0.D26	I/O3	入力	Z	1	261
□ 20	Dev0.D25	I/O3	入力	Z	1	260
□ 22	Dev0.D24	I/O3	入力	Z	1	259
□ 23	Dev0.D23	I/O3	入力	Z	1	258
□ 24	Dev0.D22	I/O3	入力	Z	1	257
□ 25	Dev0.D21	I/O3	入力	Z	1	256

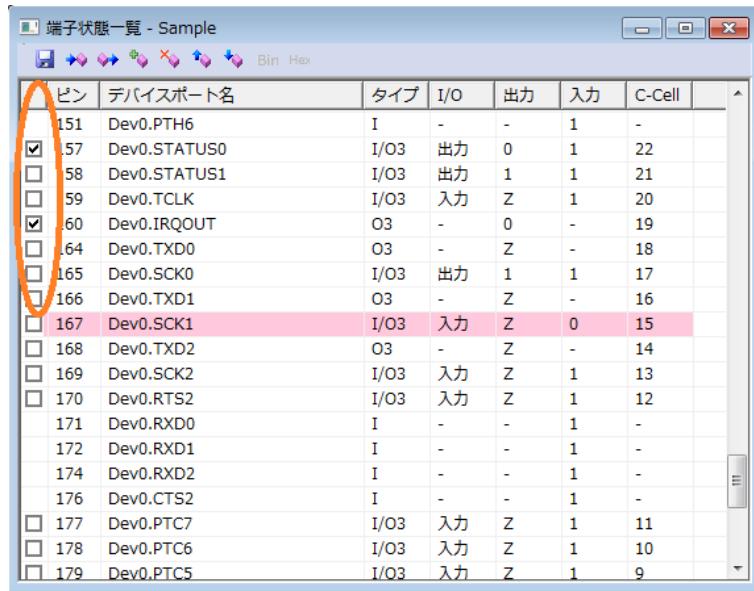
Figure 38

2.4.10. EXTEST コマンド実行



バウンダリスキャンチェーン上の各デバイスに EXTEST 命令を与えて、各デバイスの入出力ピンの接続状態を確認します。EXTEST 命令実行後に本ソフトウェアが各デバイスの入出力ピンの状態を取り込み、端子状態一覧ウィンドウのリストを更新します。更新の前後で値が変わった行は、表示色が変更されます。
詳細な手順は以下の通りです。

- 1) 値を設定したいポートやグループにチェックを付けます。
 - 入力ポート（タイプカラムが "I"）にはチェックボックスはありません。
出力ポート（タイプカラムが "O"）はチェックを付けることで出力値を設定できます。
入出力ポート（タイプカラムが "I/O"）はチェックを付けることで入出力方向と出力値を設定できます。
 - チェックを切り替えたポートと同じ C-Cell 値を持つポートがある場合、そのポートのチェックも同時に切り替わります。
グループは構成するポートの中に出力ポートか入出力ポートが一つ以上あればチェックボックスが表示されます。
チェックを切り替えると、構成する全てのポートのチェック状態が切り替わります。
そして、チェックが切り替わったポートと同じ C-Cell 値を持つポートのチェックも同時に切り替わります。

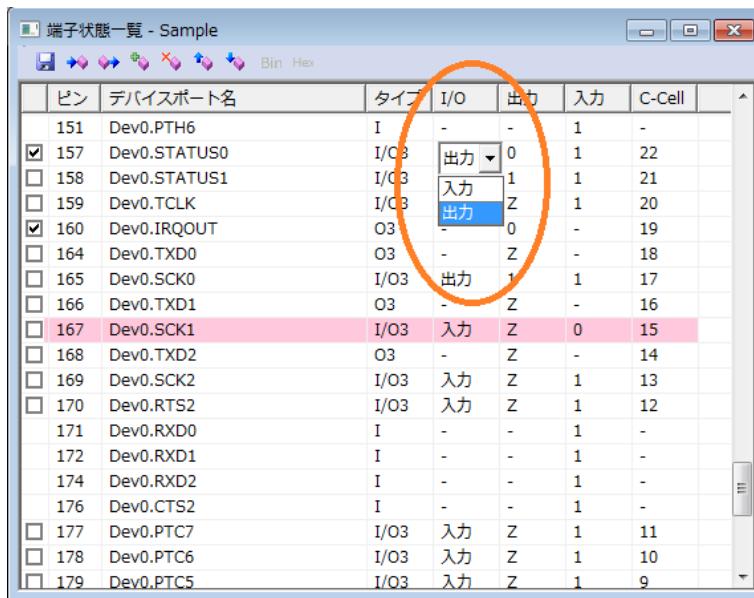


ピン	デバイスポート名	タイプ	I/O	出力	入力	C-Cell
151	Dev0.PTH6	I	-	-	1	-
<input checked="" type="checkbox"/> 157	Dev0.STATUS0	I/O3	出力	0	1	22
<input type="checkbox"/> 158	Dev0.STATUS1	I/O3	出力	1	1	21
<input type="checkbox"/> 159	Dev0.TCLK	I/O3	入力	Z	1	20
<input checked="" type="checkbox"/> 160	Dev0.IRQOUT	O3	-	0	-	19
<input type="checkbox"/> 164	Dev0.TXD0	O3	-	Z	-	18
<input type="checkbox"/> 165	Dev0.SCK0	I/O3	出力	1	1	17
<input type="checkbox"/> 166	Dev0.TXD1	O3	-	Z	-	16
<input type="checkbox"/> 167	Dev0.SCK1	I/O3	入力	Z	0	15
<input type="checkbox"/> 168	Dev0.TXD2	O3	-	Z	-	14
<input type="checkbox"/> 169	Dev0.SCK2	I/O3	入力	Z	1	13
<input type="checkbox"/> 170	Dev0.RTS2	I/O3	入力	Z	1	12
171	Dev0.RXD0	I	-	-	1	-
172	Dev0.RXD1	I	-	-	1	-
174	Dev0.RXD2	I	-	-	1	-
176	Dev0.CTS2	I	-	-	1	-
<input type="checkbox"/> 177	Dev0.PTC7	I/O3	入力	Z	1	11
<input type="checkbox"/> 178	Dev0.PTC6	I/O3	入力	Z	1	10
<input type="checkbox"/> 179	Dev0.PTC5	I/O3	入力	Z	1	9

Figure 39

2) I/O カラムをクリックして入出力方向を選択します（入出力ポートのみ）

- 入力を選択した場合は入力ポートとして使用します。
出力値の変更は行いません。
- 出力を選択した場合は 3：出力値を設定します。
- 入出力方向を切り替えたポートと同じ C-Cell 値を持つポートがある場合、そのポートの入出力方向も同時に切り替わります。



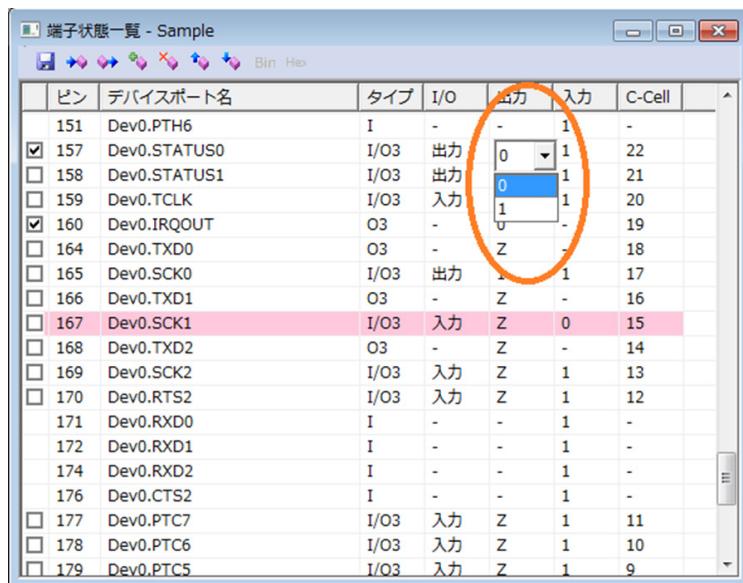
ピン	デバイスポート名	タイプ	I/O	出力	入力	C-Cell
151	Dev0.PTH6	I	-	-	1	-
<input checked="" type="checkbox"/> 157	Dev0.STATUS0	I/O3	出力	0	1	22
<input type="checkbox"/> 158	Dev0.STATUS1	I/O3	入力	1	1	21
<input type="checkbox"/> 159	Dev0.TCLK	I/O3	出力	Z	1	20
<input checked="" type="checkbox"/> 160	Dev0.IRQOUT	O3	-	0	-	19
<input type="checkbox"/> 164	Dev0.TXD0	O3	-	Z	-	18
<input type="checkbox"/> 165	Dev0.SCK0	I/O3	出力	1	1	17
<input type="checkbox"/> 166	Dev0.TXD1	O3	-	Z	-	16
<input type="checkbox"/> 167	Dev0.SCK1	I/O3	入力	Z	0	15
<input type="checkbox"/> 168	Dev0.TXD2	O3	-	Z	-	14
<input type="checkbox"/> 169	Dev0.SCK2	I/O3	入力	Z	1	13
<input type="checkbox"/> 170	Dev0.RTS2	I/O3	入力	Z	1	12
171	Dev0.RXD0	I	-	-	1	-
172	Dev0.RXD1	I	-	-	1	-
174	Dev0.RXD2	I	-	-	1	-
176	Dev0.CTS2	I	-	-	1	-
<input type="checkbox"/> 177	Dev0.PTC7	I/O3	入力	Z	1	11
<input type="checkbox"/> 178	Dev0.PTC6	I/O3	入力	Z	1	10
<input type="checkbox"/> 179	Dev0.PTC5	I/O3	入力	Z	1	9

Figure 40

3) 出力カラムをクリックして出力値を選択します。

- コントロールセルを持つ出力専用ポートは "Z" を選択して出力禁止にすることができます。
それ以外のポートは 0(Low 出力)か 1(High 出力)を選択します。
- 出力値が "0" または "1" から "Z" に切り替わるとき、もしくは "Z" から他の値に切り替わるとき、そのポートと同じ C-Cell 値を持つポートがあれば同時に値が切り替わります。
- グループの場合はエディットボックスによりグループ値を設定します。

ただし、グループを構成するポートの種類や入出力状態によっては値が無視されるビットがあります。
グループ値を2進数で設定するか16進数で設定するかはグループの現在の表示フォーマットに従います。



Pin	デバイスポート名	Type	I/O	出力	入力	C-Cell
151	Dev0.PTH6	I	-	1	-	
<input checked="" type="checkbox"/> 157	Dev0.STATUS0	I/O3	出力	0	1	22
<input type="checkbox"/> 158	Dev0.STATUS1	I/O3	出力	0	1	21
<input type="checkbox"/> 159	Dev0.TCLK	I/O3	入力	1	1	20
<input checked="" type="checkbox"/> 160	Dev0.IRQOUT	O3	-	0	-	19
<input type="checkbox"/> 164	Dev0.TXD0	O3	-	Z	-	18
<input type="checkbox"/> 165	Dev0.SCK0	I/O3	出力	1	1	17
<input type="checkbox"/> 166	Dev0.TXD1	O3	-	Z	-	16
<input type="checkbox"/> 167	Dev0.SCK1	I/O3	入力	Z	0	15
<input type="checkbox"/> 168	Dev0.TXD2	O3	-	Z	-	14
<input type="checkbox"/> 169	Dev0.SCK2	I/O3	入力	Z	1	13
<input type="checkbox"/> 170	Dev0.RTS2	I/O3	入力	Z	1	12
171	Dev0.RXD0	I	-	-	1	-
172	Dev0.RXD1	I	-	-	1	-
174	Dev0.RXD2	I	-	-	1	-
176	Dev0.CTS2	I	-	-	1	-
<input type="checkbox"/> 177	Dev0.PTC7	I/O3	入力	Z	1	11
<input type="checkbox"/> 178	Dev0.PTC6	I/O3	入力	Z	1	10
<input type="checkbox"/> 179	Dev0.PTC5	I/O3	入力	Z	1	9

Figure 41

4) 値を指定したい全てのポートに対して1～3を繰り返します。

- チェックがない場合、各ポートは以下の動作になります。
 - ・ 入出力ポート >> 入力ポートとして使用
 - ・ 出力ポート(C-Cellあり) >> 出力禁止状態で使用
 - ・ 出力ポート(C-Cellなし) >> 0(Low)を出力

5) EXTEST 実行

- EXTEST 実行時に"この設定で EXTEST を実行してよいですか"の旨の警告メッセージが表示されます。実行してよければ "OK"を、設定を見直す場合は"キャンセル"を選択してください。この警告メッセージを表示しない場合は"今後、このダイアログボックスを表示しない"にチェックを付けてください。このチェックは、"端子状態一覧"メニューからでも設定できます。

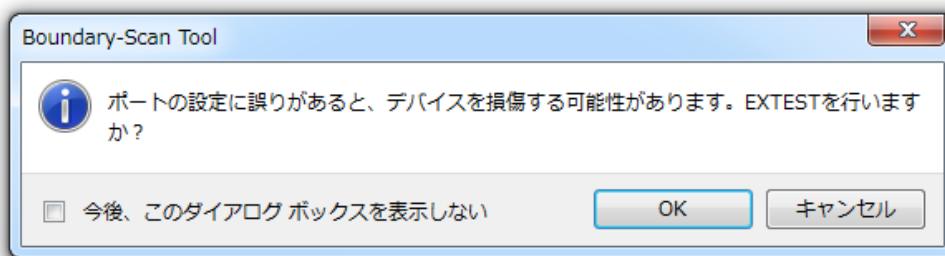


Figure 42

! ポートの入出力方向や出力値によってはターゲットボードに予期せぬ負荷がかかりボードが破損する恐れがあります。
設定に問題がないか、よくご確認のうえでコマンドを実行してください。

 **"デバイスのプロパティ"**でバイパス設定にされているデバイスのポートは、各コマンドの実行時にバウンダリスキャンのBYPASS命令によりスキップされるため、コマンドの実行前後で値が変化しません。
またそのようなポートはグレーの背景色で表示されます。

2.4.11. CSV ファイルに保存

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 端子状態一覧を CSV ファイルに保存 *端子状態一覧ウィンドウ → ツールバー → “CSV 保存” 選択	---

上記メニューを選択するとファイルの保存ダイアログが表示されますので、保存場所とファイル名を指定してください。ファイル名の初期値は端子状態一覧リストの名前になります。保存をクリックすることで下図のようなファイルが作成されます。

	A	B	C	D	E	F	G	H	I
112	DISABLE	151	Dev0.PTH6I	-	-		1	-	
113	TRUE	157	Dev0.STATI/03	出力		0	1	22	
114	FALSE	158	Dev0.STATI/03	出力		1	1	21	
115	FALSE	159	Dev0.TCLK1/03	入力	Z		1	20	
116	TRUE	160	Dev0.IRQC03	-		0	-	19	
117	FALSE	164	Dev0.TXDC03	-	Z	-		18	
118	FALSE	165	Dev0.SCKCI/03	出力		1	1	17	
119	FALSE	166	Dev0.TXD1/03	-	Z	-		16	
120	FALSE	167	Dev0.SCK1I/03	入力	Z		1	15	
121	FALSE	168	Dev0.TXD2/03	-	Z	-		14	
122	FALSE	169	Dev0.SCK2I/03	入力	Z		1	13	
123	FALSE	170	Dev0.RTS2I/03	入力	Z		1	12	
124	DISABLE	171	Dev0.RXD0I	-	-		1	-	
125	DISABLE	172	Dev0.RXD1I	-	-		1	-	
126	DISABLE	174	Dev0.RXD2I	-	-		1	-	
127	DISABLE	176	Dev0.CTS2I	-	-		1	-	
128	FALSE	177	Dev0.PTC7I/03	入力	Z		1	11	
129	FALSE	178	Dev0.PTC6I/03	入力	Z		1	10	
130	FALSE	179	Dev0.PTC5I/03	入力	Z		1	9	
131	FALSE	180	Dev0.PTC4I/03	入力	Z		1	8	
132	FALSE	182	Dev0.WAKEI/03	入力	Z		1	7	
133	FALSE	184	Dev0.RESEI/03	出力		1	1	6	
134	FALSE	185	Dev0.PTC3I/03	入力	Z		1	5	
135	FALSE	186	Dev0.PTC2I/03	入力	Z		1	4	
136	FALSE	187	Dev0.PTC1I/03	入力	Z		1	3	
137	FALSE	188	Dev0.PTC0I/03	入力	Z		1	2	
138	FALSE	189	Dev0.DRAFI/03	入力	Z		1	1	
139	FALSE	190	Dev0.DRAFI/03	入力	Z		1	0	
140	DISABLE	191	Dev0.DREGI	-	-		1	-	
141	DISABLE	192	Dev0.DREGI	-	-		1	-	
142	DISABLE	195	Dev0.MDRI	-	-		0	-	

Figure 43

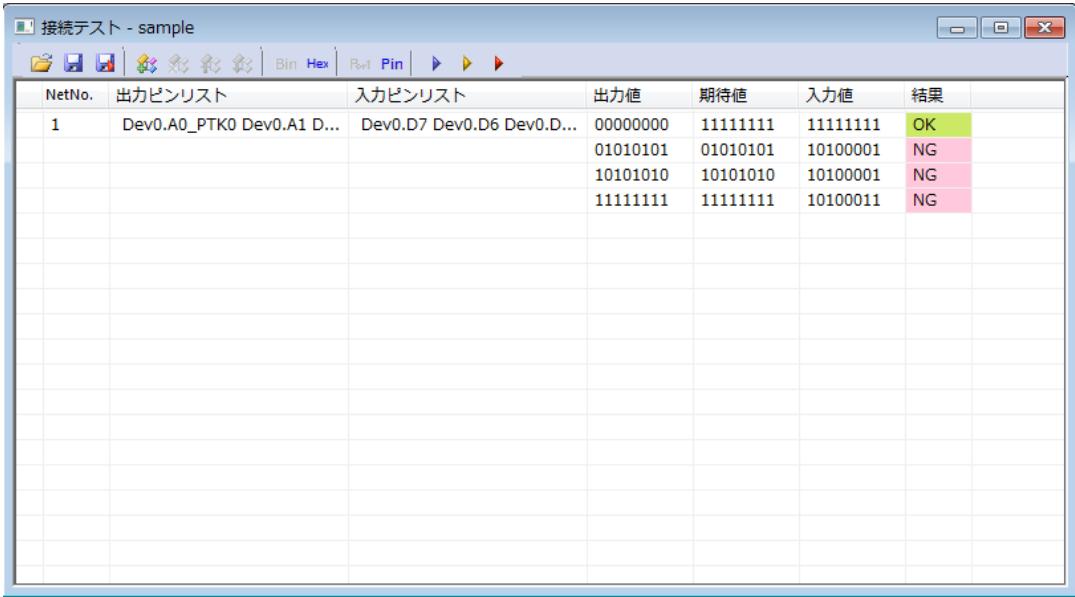
- CSV ファイルの項目は左から順に
"チェックボックス"・"ピン"・"デバイスピート名"・"タイプ"・"I/O"・"出力"・"入力"・"C-Cell"
に対応しています。
- グループ行は"C-Cell"の隣にグループを構成しているポートのピン番号が出力されます。
- チェックボックスの状態は
 - チェックボックスなし : DISABLE
 - チェック ON : TRUE
 - チェック OFF : FALSE

で表されます。その他のカラムは端子状態一覧ウィンドウの現在値が表示されます。



2.5. 接続テストウィンドウ

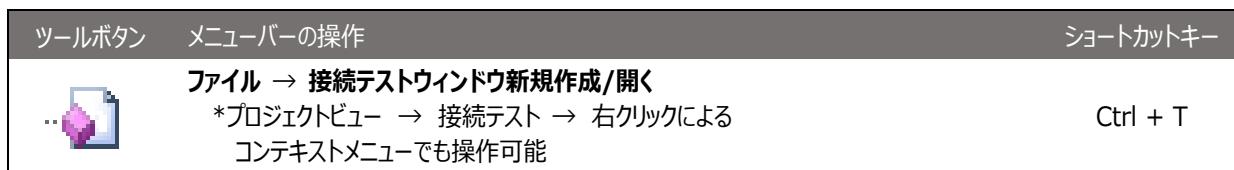
この節では本ソフトウェアの接続テストウィンドウの操作方法について説明します。



NetNo.	出力ピンリスト	入力ピンリスト	出力値	期待値	入力値	結果
1	Dev0.A0_PTK0 Dev0.A1 D...	Dev0.D7 Dev0.D6 Dev0.D...	00000000	11111111	11111111	OK
			01010101	01010101	10100001	NG
			10101010	10101010	10100001	NG
			11111111	11111111	10100011	NG

Figure 44

2.5.1. 接続テストウィンドウの新規作成/開く



上記メニューを実行すると、プロジェクトビューに"test n"が追加されるとともに接続テストウィンドウが開きます。（n は 1 からの通し番号） 上記メニューを実行するたびに、新たに空の"test n" が追加され、追加された空のリストがウィンドウに表示されます。ただし、プロジェクトビュー上で"test n" を選択した状態で上記メニューを実行すると、新たなリストは作成せず、選択したリストをウィンドウに表示します。現在開いているリストは接続テストウィンドウのタイトルバーに表示されます。

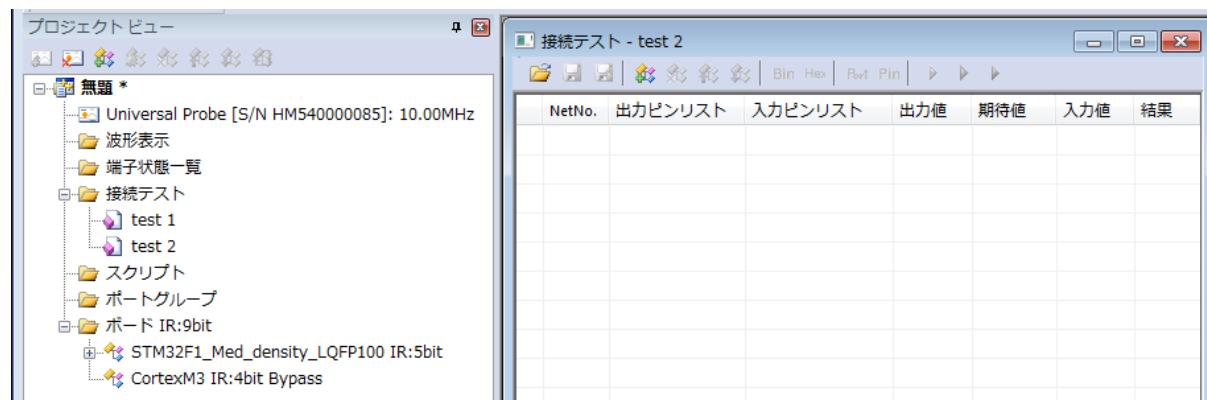
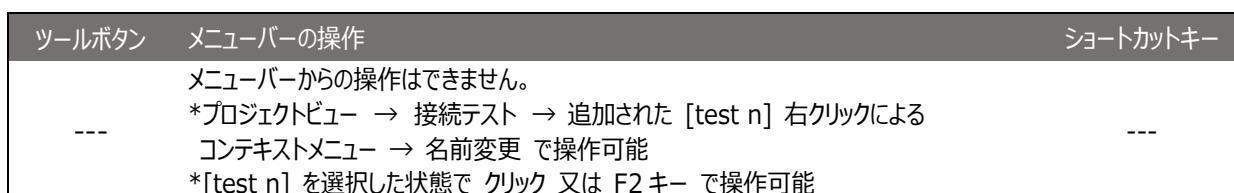


Figure 45

2.5.2. テスト（リスト）の名前変更



上記メニューを選択すると、プロジェクトビューの該当項目が編集状態になりますので、目的に合わせて名称を変更してください。

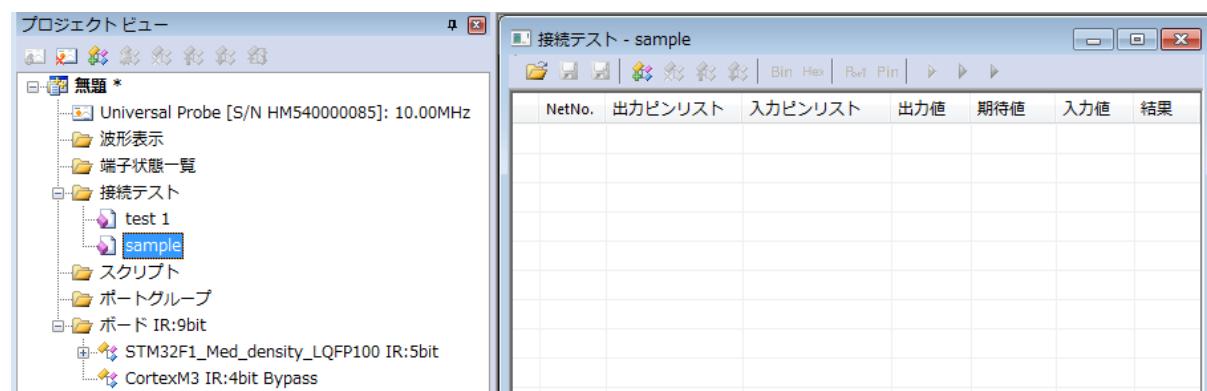
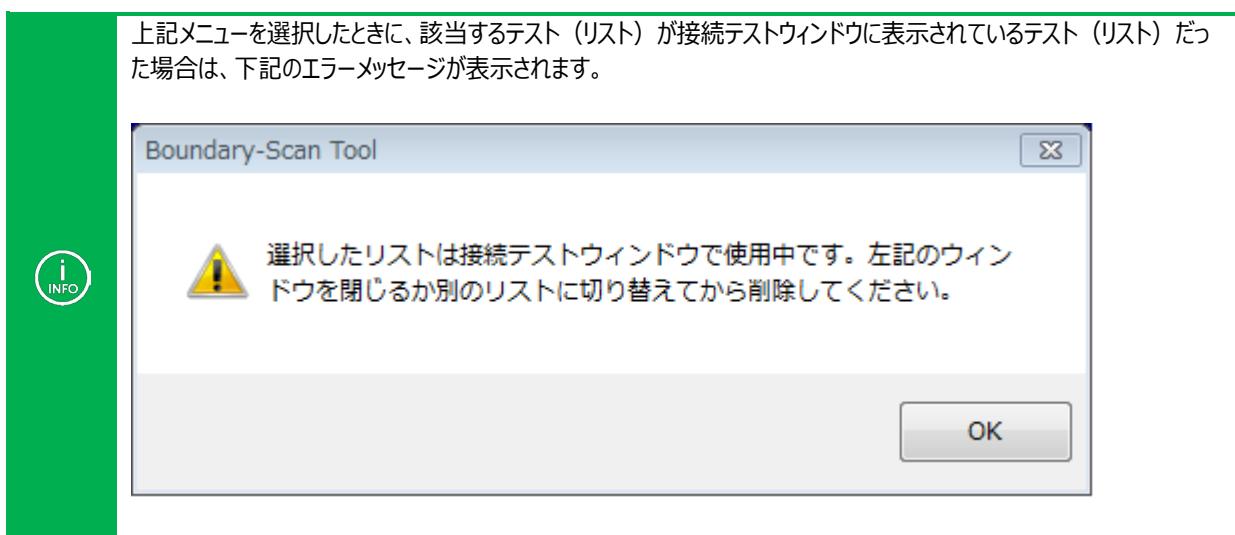


Figure 46

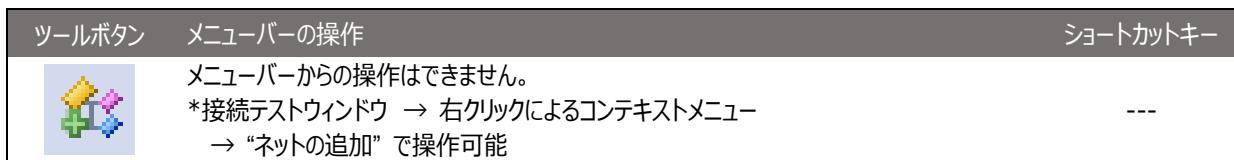
2.5.3. テスト（リスト）の削除

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *プロジェクトビュー → 接続テスト → 追加された [test n] 右クリックによる コンテキストメニュー → 削除 で操作可能	---

上記メニューを選択すると、プロジェクトビューの該当項目が削除されます。



2.5.4. ネットの追加



上記メニューを選択すると空の"net n"行が接続テストウィンドウに追加されます。（n は 1 からの通し番号）

NetNo.	出力ピンリスト	入力ピンリスト	出力値	期待値	入力値	結果
1						

出力ピンが設定されていません。

Figure 47



背景がピンク色のカラムは詳細設定に不備があるカラムです。マウスカーソルを合わせると不備の理由がポップアップ表示されます。各カラムは不備がなくなった時点で背景が青紫色に変わります。

2.5.5. ピンリストの編集

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 *接続テストウィンドウ → 入力/出力ピンリスト → 右クリックによるコンテキストメニュー → “入力/出力ピンリストの編集”で操作可能	---

上記メニューを選択すると“ピンリストの編集”ダイアログが開きます。接続テストを実行したいピンのリストを構成してください。

- ・**ピンリストの編集ダイアログ**

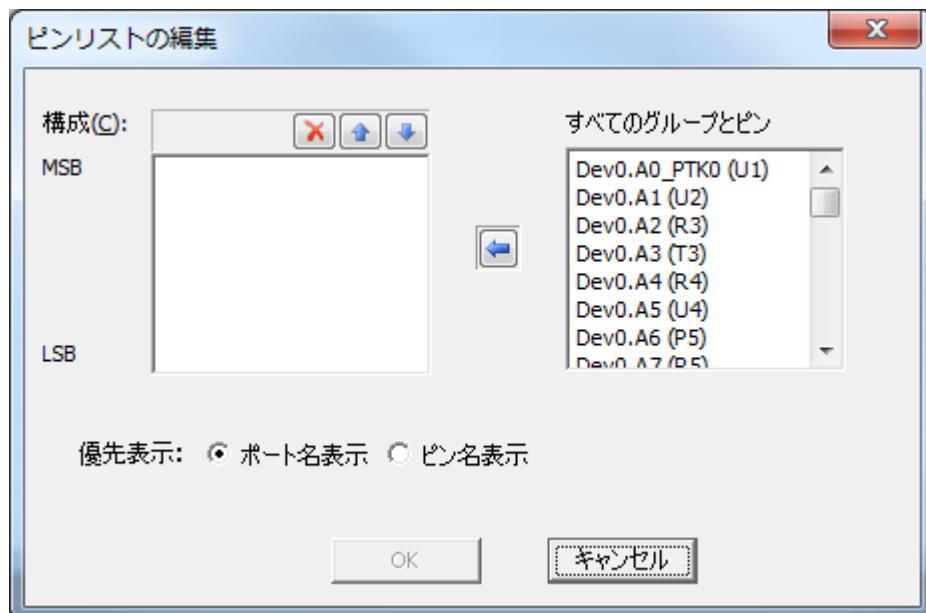


Figure 48

- ・**構成するポートを指定する**

「すべてのグループとピン」リストにボード上のすべてのピンや追加したグループがリストされます。グループに追加したいピンやグループをそのリストで選択し[←]ボタンをクリックして追加します。「構成」リストの最後に追加されます。

- ・**構成するポートの順序を変える**

「構成」リストで順番を変更したいピン/グループを選択し[↑]ボタンまたは[↓]ボタンをクリックして順番を変更します。

- ・**構成するポートを削除する**

「構成」リストから削除したいピン/グループを選択し[×]ボタンをクリックして削除します。

- ・**名前の表示方法を変更する**

「ポート名表示」または「ピン名表示」をクリックすることで入力/出力ピンリストカラムでの表示が切り替わります。

- ・**編集を反映する**

[OK]ボタンをクリックしてピンリストの編集を反映させます。

 出力ピンリストに入力専用ピンが含まれていると、設定不備として出力ピンリストカラムの背景色がピンク色に変わります。

同様に入力ピンリストに出力専用ピンが含まれていると、設定不備として入力ピンリストカラムの背景色がピンク色に変わります。

その他の編集手段として、下記の方法があります。

- プロジェクトビューのピンやグループを出/入力ピンリストカラムにドラッグ&ドロップをすると、そのピンやグループが登録されます。
- 出/入力ピンリストカラム上でマウスの左ボタンをクリックすると編集用のエディットボックスが開きます。リストを直接編集してください。ピン名の区切りはスペース文字です。

リストに不備がなければ、編集したカラムの背景色が青紫色に変わります。

また、入力/出力ピンリストのカラムにマウスカーソルを合わせると、全ての構成要素をポップアップ表示します。

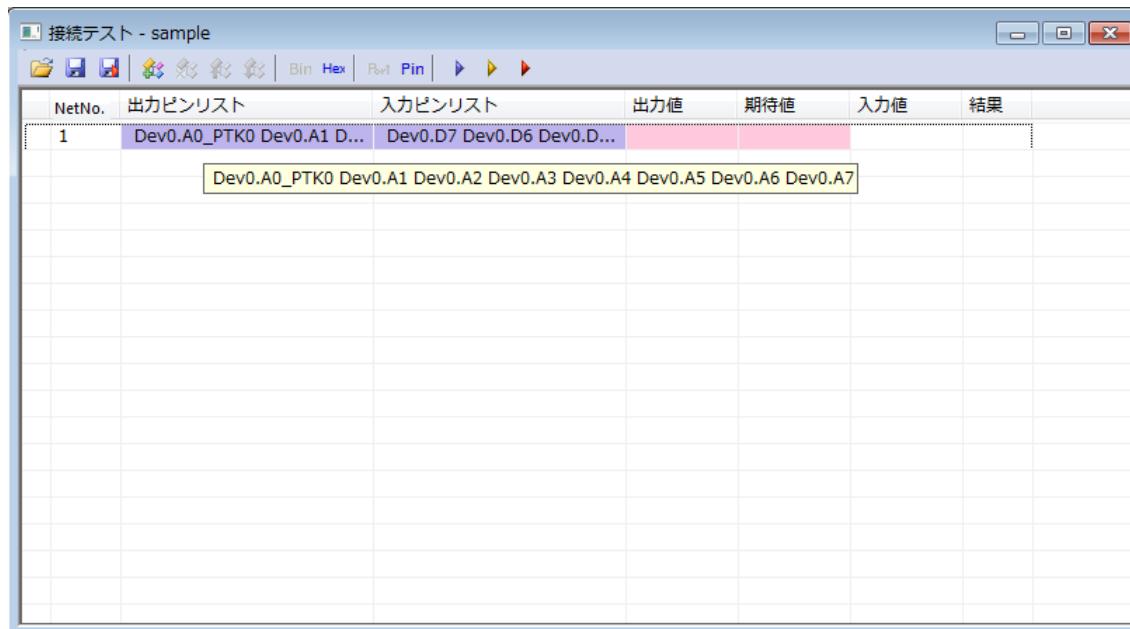


Figure 49

2.5.6. ネットの削除

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 右クリックによるコンテキストメニュー → “ネットの削除”で操作可能	---

上記メニューを選択すると“net n”行が削除されます。出力値パターンが複数登録されている場合はその行も同時に削除されます。

2.5.7. ネットの表示順序変更

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 右クリックによるコンテキストメニュー → “ネットの上移動”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 右クリックによるコンテキストメニュー → “ネットの下移動”で操作可能	---

"ネットの上移動"を実行すると、選択した"net n"が接続テストのリスト上でひとつ上のものに入れ替わります。
 "ネットの下移動"を実行すると、選択した"net n"が接続テストのリスト上でひとつ下のものに入れ替わります。

2.5.8. ネットリストを開く

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → ネットリストを開く	Ctrl+Shift+T

上記メニューを選択するとファイル選択ダイアログが開きます。ネットリストファイル（bnt ファイル）を選択してください。ネットリストファイルはフォーマットに従ってあらかじめ作成しておくか、このウィンドウ上で構築したものを保存して作成します。

	<ul style="list-style-type: none"> プロジェクトビューの『接続テスト』タグの下に登録されているテスト名と同名のファイルを開いた場合はそのテストにネットが再構築されます。 違う名前のファイルを開いた場合はプロジェクトビューのテスト名がファイルのテスト名に変更された上で、ファイルに記述されているネットが再構築されます。 ネットリストファイルに不備があった場合、エラーリストファイル（ber ファイル）がネットリストファイルのフォルダーに作成されます。ネットリストファイルの修正に役立ててください。
---	---

2.5.9. グループの作成/編集/削除

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューバーからの操作はできません。 --- *接続テストウィンドウ → 右クリックによる コンテキストメニュー → "グループ作成/編集/削除" で操作可能	---

上記メニューを選択するとグループ管理ダイアログが開きます。このダイアログで接続テストウィンドウ内にて使えるグループの作成・編集・削除を行うことができます。

・グループ管理ダイアログ

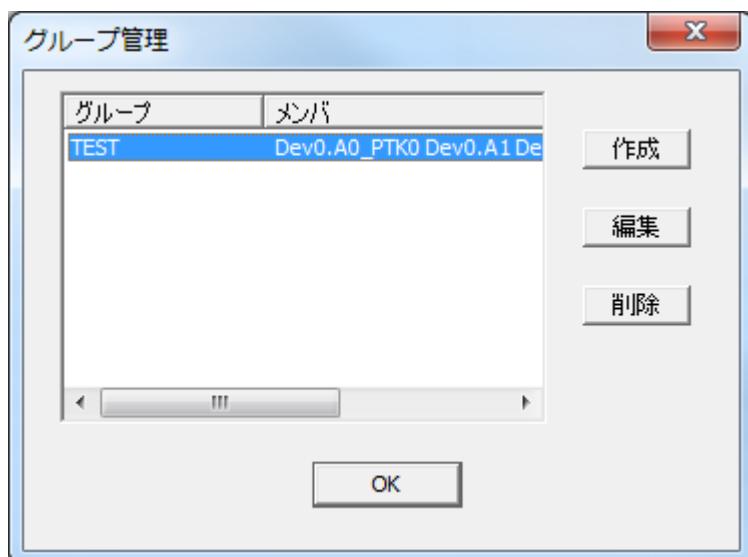


Figure 50

作成ボタン及び編集ボタンをクリックするとさらにグループ作成ダイアログが開きます。ボード上の全ポートの組み合わせを指定して、グループを作成してください。

このメニューで作成したグループはプロジェクトビューには追加されません。

グループ管理ダイアログにはプロジェクトビューで作成したグループも表示されます。そのグループを編集した場合、変更点は接続テストウィンドウ内でのみ有効です。プロジェクトビューの方には反映されません。また、削除はできません。

・グループ作成ダイアログ

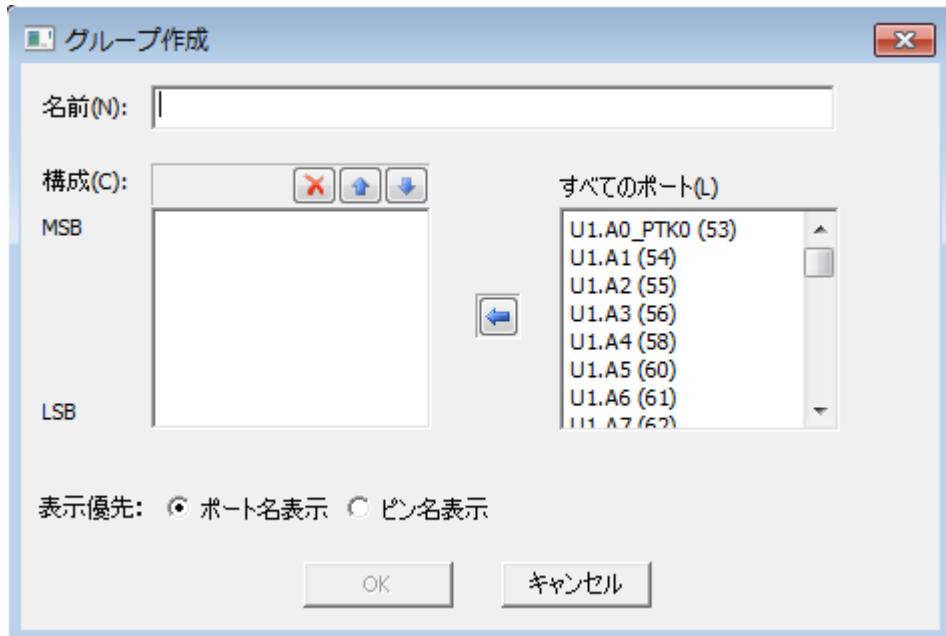


Figure 51

・名前を指定する

「名前」グループ名を指定します

・構成するポートを指定する

「すべてのポート」リストにボード上のすべてのポートがリストされます。グループに追加したいポートをそのリストで選択し[←]ボタンをクリックしてグループに追加します。「構成」リストの最後に追加されます。

・構成するポートの順序を変える

「構成」リストで順番を変更したいポートを選択し[↑]ボタンまたは[↓]ボタンをクリックしてポートの順番を変更します。

・構成するポートを削除する

「構成」リストで削除したいポートを選択し[×]ボタンをクリックしてポートを削除します。

・ピンを表示するフォーマットを指定する

「ポート名表示」または「ピン名表示」をクリックすることでグループの構成の表示するフォーマットを指定できます。

・グループを追加する

[OK]ボタンをクリックしてグループを追加します。追加されたグループは出/入力ピンリストの編集時に選択することができます。

2.5.10. ピン/ポート名表示切り替え

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 入力/出力ピンリスト → 右クリックによる コンテキストメニュー → “ポート名表示”で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 入力/出力ピンリスト → 右クリックによる コンテキストメニュー → “ピン名表示”で操作可能	---

選択したカラムのピン表示を ポート名または ピン名で表示するか切り替えます。
 もしくは、出力ピンリストカラムまたは入力ピンリストカラムをどこも選択していない状態でツールバーのボタンを押すことで、全てのピンの表示フォーマットを切り替えることができます。

2.5.11. 出力値の設定

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 出力ピンリスト → 右クリックによる コンテキストメニュー → “出力パターンの自動生成”で操作可能 *接続テストウィンドウ → 出力値 → 右クリックによる コンテキストメニュー → “出力パターンの追加/削除”で操作可能	---

接続テスト実行時に出力ピンリストの各端子から出力する値を設定します。

出力パターンの自動生成ボタンをクリックすると、「全端子を 0」・「全端子を 1」・「0101...」・「1010...」の 4 通りの出力パターンを挿入します。

それ以外の出力値を設定したい場合は、自動生成で作成されたパターンをクリックして直接編集するか『出力パターンの追加』メニューにより新しく出力パターン行を追加して編集してください。

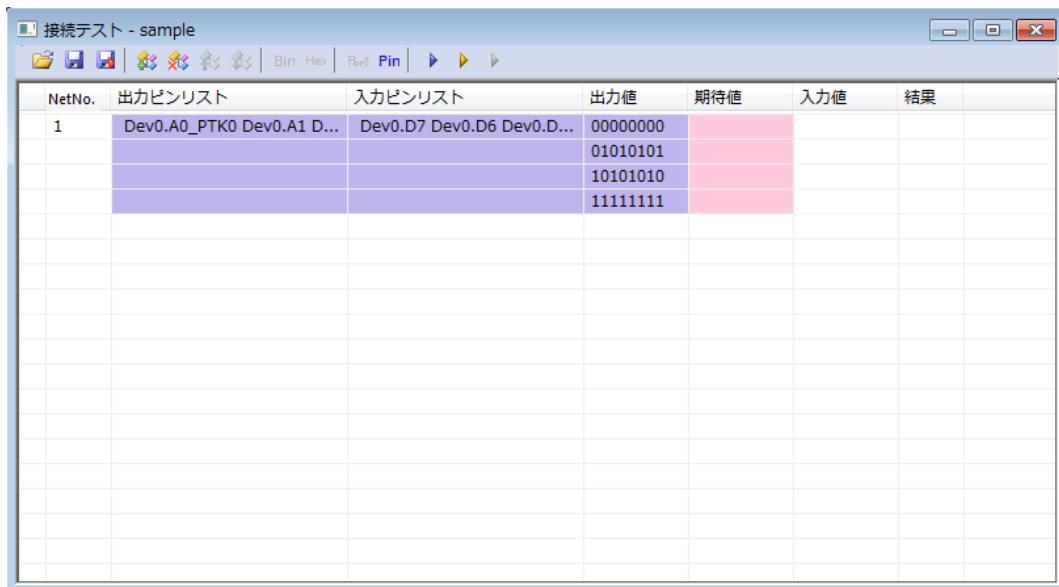
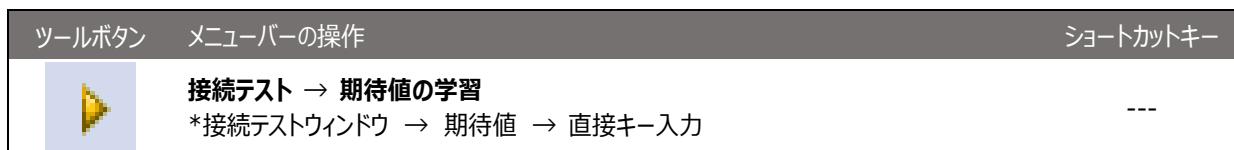


Figure 52

2.5.12. 期待値の設定

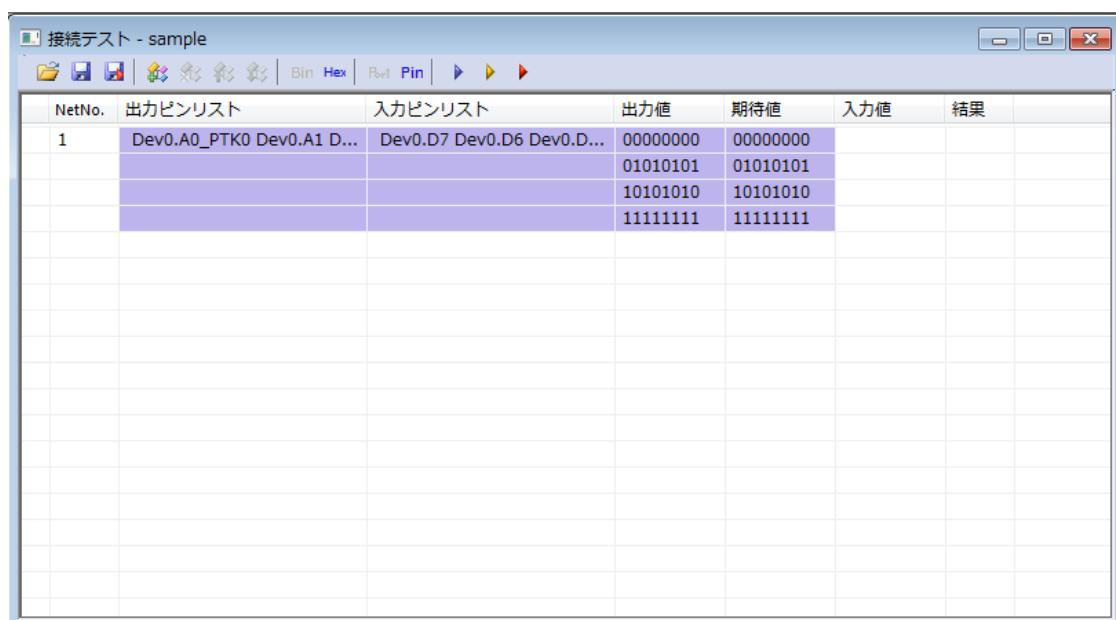


設定した出力値で接続テストを行った際に入力ピンリストの各端子が取るはずの値を設定します。
 文字"x"を記述した場合、そのビットはテスト時に入力値との比較を行いません（常にOKとして扱います）。
 16進数表示をしているときの文字"x"は「4ビットまとめて無視」を意味します。

『期待値の学習』メニューを実行すると、バウンダリスキャンテストの EXTEST コマンドを実行して実際に値を取り込み、それを期待値にします。
 既に接続テストをパスしているボードがお手元にある場合はそのボードを利用して期待値を生成することができます。

期待値の実行対象は、以下の通りです。

- ネット番号カラムをクリックして行を選択した場合、選択したネットの全パターンが実行の対象となります。
- それ以外のカラム（出力ピンリスト、入力ピンリスト、出力値、期待値、入力値、結果）をクリックして行を選択した場合、選択した行のみが実行の対象となります。
- 出力ピンリスト、入力ピンリスト、出力値、期待値のカラムが編集状態、若しくは、設定を編集した直後は、その行のみが実行の対象となります。
- ネットが登録されていない行をクリックする等をして、どの行も選択状態になっていない場合、登録されているネット全体が実行対象となります。



NetNo.	出力ピンリスト	入力ピンリスト	出力値	期待値	入力値	結果
1	Dev0.A0_PTK0 Dev0.A1 D...	Dev0.D7 Dev0.D6 Dev0.D...	00000000	00000000		
			01010101	01010101		
			10101010	10101010		
			11111111	11111111		

Figure 53

2.5.13. テストの実行



バウンダリスキャンテストの EXTEST コマンドを使用して接続テストを実行します。
出力値と期待値が設定されている行はテストを行うことができます。

接続テストの対象は、以下の通りです。

- ・ネット番号カラムをクリックして行を選択した場合、選択したネットの全パターンが実行の対象となります。
- ・それ以外のカラム（出力ピンリスト、入力ピンリスト、出力値、期待値、入力値、結果）をクリックして行を選択した場合、選択した行のみが実行の対象となります。
- ・出力ピンリスト、入力ピンリスト、出力値、期待値のカラムが編集状態、若しくは、設定を編集した直後は、その行のみが実行の対象となります。
- ・ネットが登録されていない行をクリックする等をして、どの行も選択状態になっていない場合、登録されているネット全体が実行対象となります。

出力テストでは、まず"出力ピンリスト"で指定された端子を EXTEST コマンドにより"出力値"で指定された値にドライブします。

ドライブ後、入力ピンリストで指定された端子の値を取り込み、期待値との比較を行います。

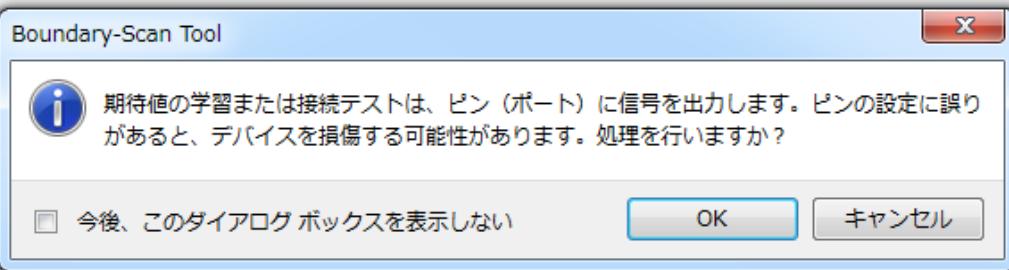
取り込んだ値と期待値が同じであれば結果カラムに「OK」が表示されて結果カラムの背景色が緑色になります。

取り込んだ値と期待値が異なっていたときは結果カラムに「NG」が表示されて結果カラムの背景色がピンク色になります。

NetNo.	出力ピンリスト	入力ピンリスト	出力値	期待値	入力値	結果
1	Dev0.A0_PTK0 Dev0.A1_D...	Dev0.D7 Dev0.D6 Dev0.D...	00000000 01010101 10101010 11111111	11111111 01010101 10101010 11111111	11111111 10100001 10100001 10100011	OK NG NG NG

Figure 54

EXTEST 実行時に"この設定で EXTEST を実行してよいか"の旨の警告メッセージが表示されます。実行してよければ"OK"を、設定を見直す場合は"キャンセル"を選択してください。この警告メッセージを表示しない場合は"今後、このダイアログボックスを表示しない"にチェックを付けてください。このチェックは、"接続テスト"メニューからでも設定できます。



後でネットを追加した場合も、上記 1~3 を繰り返してテストを行います。

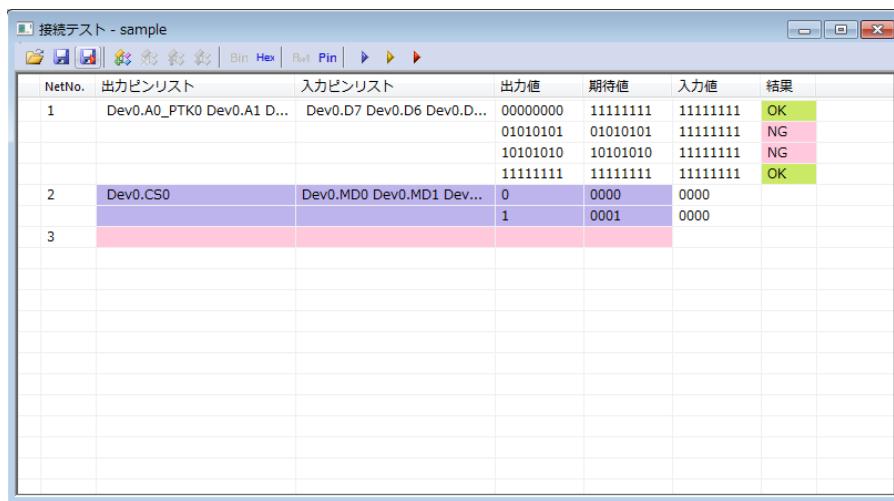


Figure 55

2.5.14. 2進/16進表示切り替え

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 出力値/期待値/入力値 → 右クリックによる コンテキストメニュー → "2進数表示" で操作可能	---

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。 *接続テストウィンドウ → 出力値/期待値/入力値 → 右クリックによる コンテキストメニュー → "16進数表示" で操作可能	---

マウスカーソルが示したカラムの出力値、期待値または入力値を2進数または16進数で表示するか切り替えます。どこも選択していない状態でツールバーのボタンを押すことでの、全ての出力値、期待値、入力値の表示フォーマットを切り替えることができます。

ただし、文字"x"を含む期待値を2進数から16進数に切り替える場合、文字"x"は4ビット（16進数の1桁分）連続している必要があります。

2.5.15. ネットリストの保存

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → ネットリストの保存	---

上記メニューを選択するとファイルの保存ダイアログが表示されますので、保存場所とファイル名を指定してください。
ファイル名の初期値は接続テストの名前になります。

ファイルは.bnt ファイルと.bnv ファイルの 2 つが作成されます。

各ファイルのフォーマットは下記の通りです。

Table 11

.bnt ファイル	
[DeviceName]	<p>デバイス参照名のリストです。 "デバイス参照名, デバイス初期名" の組み合わせです。どのデバイスもデバイス参照名の設定がされていない場合はこのタグは省略されます。</p> <p>例 :</p> <p>[DeviceName] U1, Dev0 U2, Dev1</p>
[PortNameFile]	<p>デバイス単位のポート名の定義ファイル名のリストです。 "デバイス名,ポート名の定義ファイル"の組み合わせです。デバイス名は、デバイス参照名でもデバイス初期名でも構いません。どのデバイスもポート名の変更が設定されていない場合はこのタグは省略されます。</p> <p>例 :</p> <p>[PortNameFile] Dev0, test2_Dev0.pn</p>
[GroupName]	<p>接続テストで使用されているグループのリストです。 "グループ名, ピン 1 ピン 2 ..." の組み合わせです。各ピンの間はスペースで区切れます。グループが一つも使用されていない場合はこのタグは省略されます。</p> <p>例 :</p> <p>[GroupName] TESTGROUP, U1.1 U1.2 U1.3 U1.4</p>
[Net]	<p>ネットの設定のリストです。 "ネット番号, 出力ピンリスト, 入力ピンリスト" の組み合わせです。各ピンの間はスペースで区切ります。</p> <p>例 :</p> <p>[Net] 1, U1.1 U1.2 U1.3 U1.4..., U2.1 U2.2 U2.3 U2.4... 2, TESTGROUP, U1.200 U1.201 U1.202 U1.203...</p>

Table 12

.bnv ファイル	
[NetValue]	<p>ネットごとの出力値、期待値、入力値のリストです。 "ネット番号、出力値、期待値、入力値" の組み合わせです。接続テストウィンドウ上で 16 進数表示にした値は値の最後に文字"H"がつきます。また、接続テスト時の比較を無視（常に OK）するビットは文字"x"で表現されています。</p> <p>例：</p> <pre>[NetValue] 1, 00000000, xxxxxxxx, 11101001 1, 01010101, 01010101, 11111111 1, AAH, AAH, FFH 1, FFH, xFH, FFH 2, 0, 0101, 0101 2, 1, 1010, 1010</pre>

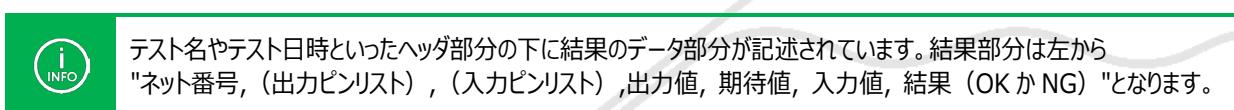
2.5.16. 接続テスト結果の保存



上記メニューを選択するとファイルの保存ダイアログが表示されますので、保存場所とファイル名を指定してください。
 ファイル名の初期値は接続テストの名前になります。ファイルフォーマットは CSV です。

接続テスト結果									
ヘッダ情報		接続テスト結果							
ネット番号	出力ピンリスト	入力ピンリスト	出力値	期待値	入力値	結果	出力値	期待値	入力値
1	U1.A0_PTK0	U1.A1 U1.A2 U1.A3 L	U1.D7 U1.D6 U1.D5 U1.D4 U1.D3	0	11100001	11101001	OK		
1			1010101	11100001	11100001	OK			
1			10101010	11100011	11100001	OK			
1			11111111	11100011	11100011	OK			
2	U1.A8 U1.A9 U1.A10 U1.A11		U1.D8 U1.D9 U1.D10 U1.D11	0	1101	1101	OK		
2				101	1101	1101	OK		
2				1010	1101	1101	OK		
2				1111	1101	1101	OK		

Figure 56



2.6. スクリプトペイン

この節では本ソフトウェアのスクリプトペインの操作方法について説明します。

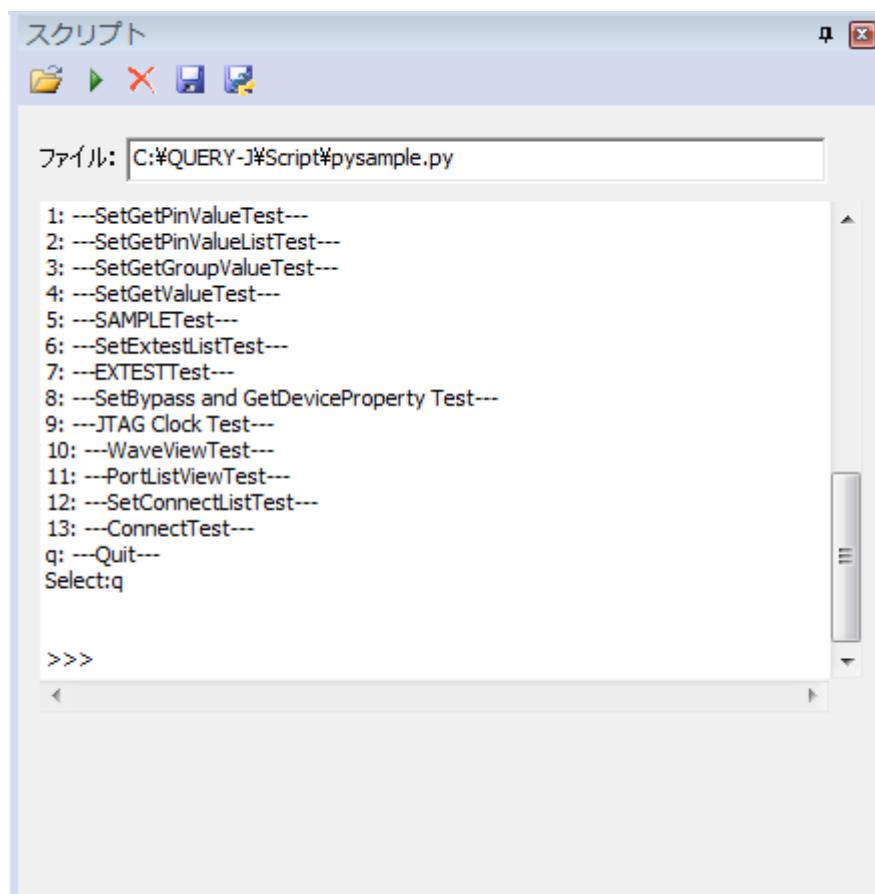
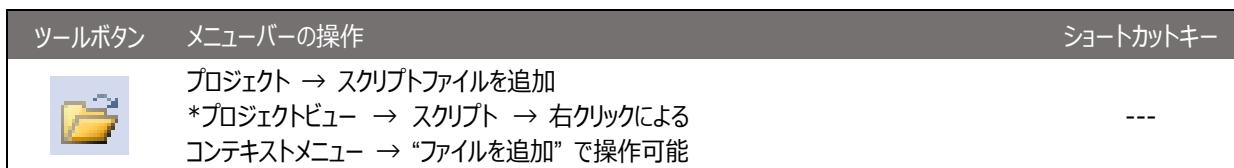


Figure 57

2.6.1. ファイル選択



スクリプトペイン上のツールバーにある上記ボタンをクリックするとファイル選択ダイアログが開きます。あらかじめ作成してある Python スクリプトファイルを選択し、ファイル選択ダイアログの **OK** ボタンを押すと 選択されたファイル名が**ファイル : エディットボックス**に表示されます。

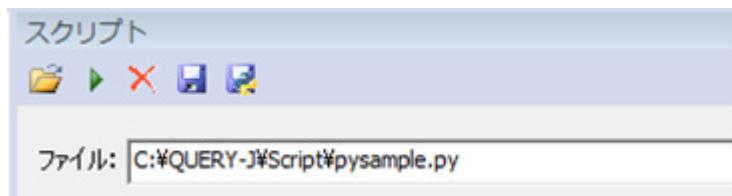


Figure 58

プロジェクトビューから操作するとファイル選択ダイアログが開きます。あらかじめ作成してある Python スクリプトファイルを選択し、ファイル選択ダイアログの OK ボタンを押すと 選択されたファイル名がプロジェクトビューのスクリプトツリーの下に追加されます。

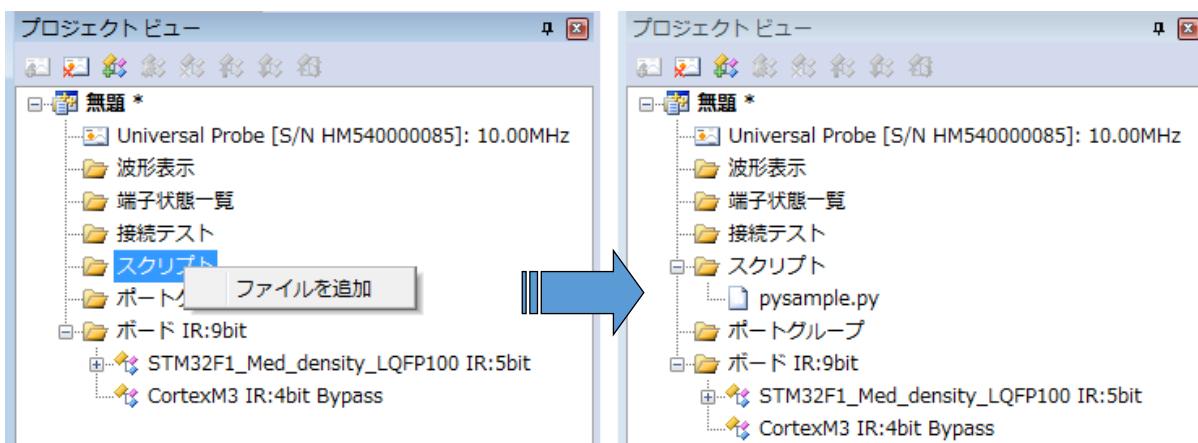


Figure 59

 スクリプトファイルの実行時に Python インタプリタ (python32.dll) を使用します。本ソフトウェアのインストール時に Python のインストールを行わなかった場合は別途 Python の実行環境をご用意ください。Python のバージョンは 3.2 を想定しています。
 本ソフトウェアに付属している Python のインストーラはバージョン 3.2.2 です。

2.6.2. 実行（ファイル）

ツールボタン	メニューバーの操作	ショートカットキー
	プロジェクト → スクリプトファイルを追加 *プロジェクトビュー → スクリプト → 右クリックによる コンテキストメニュー → “実行”で操作可能	---

スクリプトペイン上のツールバーにある上記ボタンをクリックすると**ファイル**：エディットボックスに入力されているファイルがPython インタプリタ上で実行されます。スクリプトファイルの内容が全て実行されて正常に終了した場合、実行されたファイルがプロジェクトビューの『スクリプト』ツリーに登録されなければ登録されます。実行途中でエラー終了した場合は登録されません。

	スクリプトファイルには Python3 の quit()及び exit()命令以外の記述と、本ソフトウェアが提供する 本ソフトウェアの各種機能を使用するためのインターフェース関数を使用できます。インターフェース関数についての詳細は『インターフェース関数一覧（機能別）』を参照してください。
---	--

プロジェクトビューから選択すると、同様にスクリプトファイルが Python インタプリタ上で実行されます。

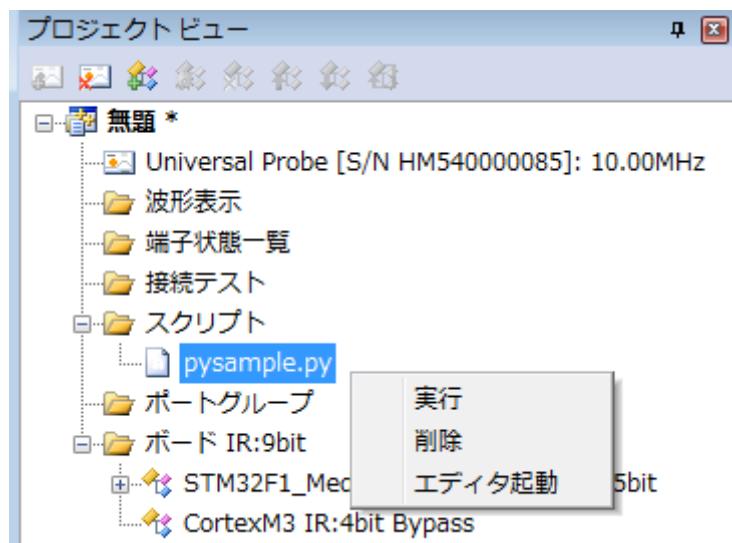
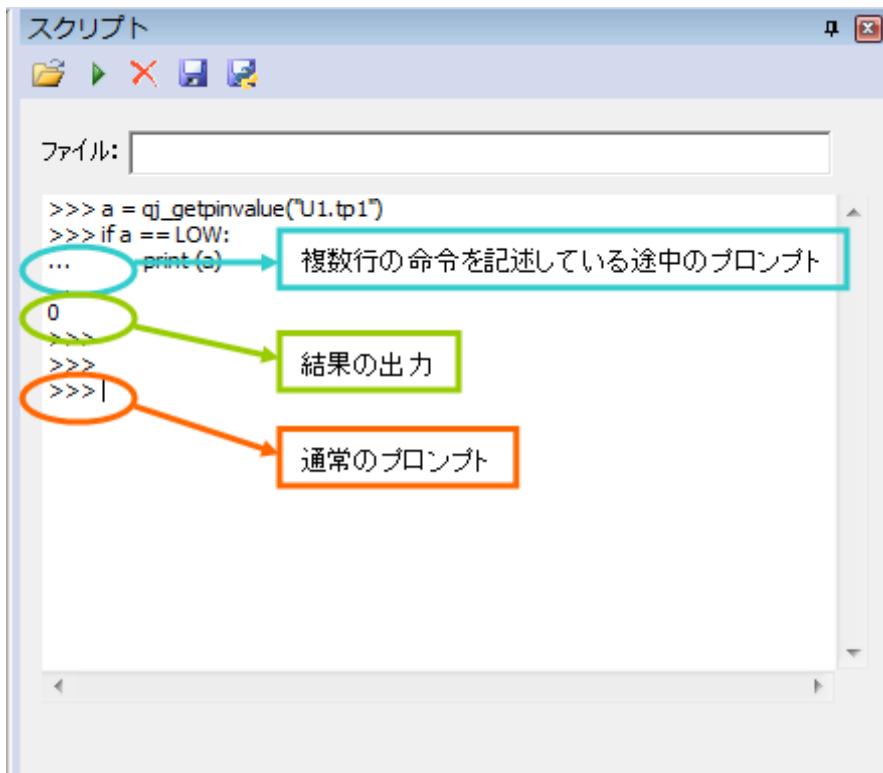


Figure 60

2.6.3. 実行（コマンドライン）



The screenshot shows a software interface titled "スクリプト" (Script). In the script editor, the following Python code is displayed:

```
>>> a = qj_getpinvalue("U1.tp1")
>>> if a == LOW:
...     print(a)
0
>>>
>>>|
```

Annotations explain the visual elements:

- A blue oval encloses the first three lines of code, with a blue arrow pointing to a yellow box labeled "複数行の命令を記述している途中のプロンプト" (Multi-line command being entered).
- A green oval encloses the number "0", with a green arrow pointing to a yellow box labeled "結果の出力" (Output result).
- An orange oval encloses the cursor position at the end of the third line, with an orange arrow pointing to a yellow box labeled "通常のプロンプト" (Normal prompt).

Figure 61

スクリプトペイン上の標準入出力領域で Python の各種コマンドを記述し、対話形式で実行することができます。本ソフトウェアを起動してから、プロジェクトを新規作成するかプロジェクトを開いてから利用可能になります。（ただし、本ソフトウェア用の各関数はターゲットと接続してから使用可能になるものがほとんどです。）通常は『>>>』のプロンプトが表示されていますので、そこにスクリプトを記述してください。if 文や for 文などの複数の行にまたがる記述の途中では『...』のプロンプトが表示されます。インデントに注意してスクリプトを記述してください。



標準入出力領域へ改行コードを含む文字列をペーストした直後はペーストした文字列は実行されません。ペースト後にエンターキーを入力した段階で実行されます。

尚、複数行の文字列をペーストしてエンターキーを押した場合に実行されるのは最下行の文字列のみになります。



コマンド実行時に Python インタプリタ (python32.dll) を使用します。本ソフトウェアのインストール時に Python のインストールを行わなかった場合は別途 Python の実行環境をご用意ください。Python のバージョンは 3.2 を想定しています。

本ソフトウェアに付属している Python のインストーラはバージョン 3.2.2 です。



スクリプトには Python3 の quit() 及び exit() 命令以外の記述と、本ソフトウェアが提供する 本ソフトウェアの各種機能を使用するためのインターフェース関数を使用できます。インターフェース関数についての詳細は『インターフェース関数一覧（機能別）』を参照してください。

2.6.4. 保存（テキスト形式）

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	---

スクリプトペイン上のツールバーにある上記ボタンをクリックするとファイル選択ダイアログが開きます。指定された名前のファイルに標準入出力エディットボックスの内容がテキスト形式で書き出されます。

2.6.5. 保存（スクリプト形式）

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	---

スクリプトペイン上のツールバーにある上記ボタンをクリックするとファイル選択ダイアログが開きます。指定された名前のファイルに標準入出力エディットボックスの内容が書き出されます。書き出す際にコマンド行からはプロンプト文字列（『>>>』・『...』）が削除されます。また実行結果の行は先頭にコメントアウト文字列（『#』）が挿入されます。書き出されたファイルを**ファイル名**：エディットボックスに記述したり、プロジェクトビューに追加することで再利用できます。

2.6.6. 表示のクリア

ツールボタン	メニューバーの操作	ショートカットキー
	メニューバーからの操作はできません。	---

スクリプトペイン上のツールバーにある上記ボタンをクリックすると標準入出力エディットボックスの内容がクリアされます。

3. メニュー

メニューについて解説します。

3.1. ファイル

「ファイル」メニューではプロジェクトに関するファイル操作を行います。

3.1.1. プロジェクトの新規作成

プロジェクトの新規作成を行います。

初めて使用する場合は、下記ツールボタンもしくはメニューを選択し、プロジェクトを作成してください。



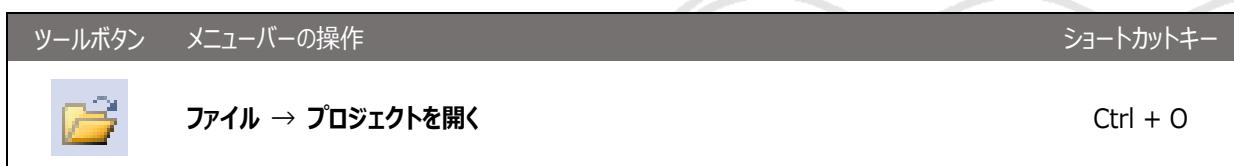
上記メニューを実行すると、プロジェクトビューに「無題」プロジェクトが作成されます。



Figure 62

3.1.2. プロジェクトを開く

以前に作成されたプロジェクトファイルを開きます。前回の作業終了時の設定が復元されます。



上記メニューを実行すると、**ファイルの選択**ダイアログが表示されます。開きたいプロジェクトファイルを選択してから **OK** を選択すると、プロジェクトファイルに記録されている情報がプロジェクトビューに復元されます。

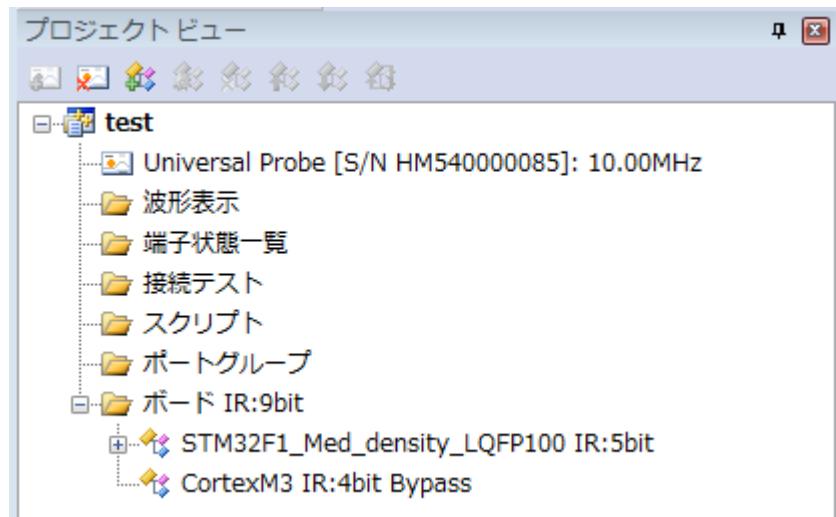


Figure 63

3.1.3. プロジェクトの保存

現在設定されている各種情報をプロジェクトファイルへ保存します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → プロジェクトの保存 ファイル → プロジェクトに名前を付けて保存	
		Ctrl + S

上記メニューを実行すると、プロジェクトの各種設定をファイルに保存します。保存するプロジェクトのファイルが存在しないときに"上書き保存"を選択するか、"名前を付けて保存"を選択したときは**ファイルの選択**ダイアログが表示されます。保存場所とファイル名を指定してから**保存**を選択すると、プロジェクトの設定がファイルに保存されます。

プロジェクトファイルの拡張子は .bss です。

プロジェクトファイルは以下の情報を記録しています。

- 接続対象となるプローブの個体 ID とバウンダリスキャンテストを行うクロック周波数 (JTAG クロック)
- バウンダリスキャンテストの対象となるターゲットのデバイス構成
- 波形表示ウィンドウに登録したポート情報 (波形表示パターン)
- 端子状態一覧ウィンドウに登録したポート情報 (端子状態一覧パターン) 及び各ポートのチェック状態
- "プロジェクト >> JTAG 信号の強制出力"メニューのチェック状態

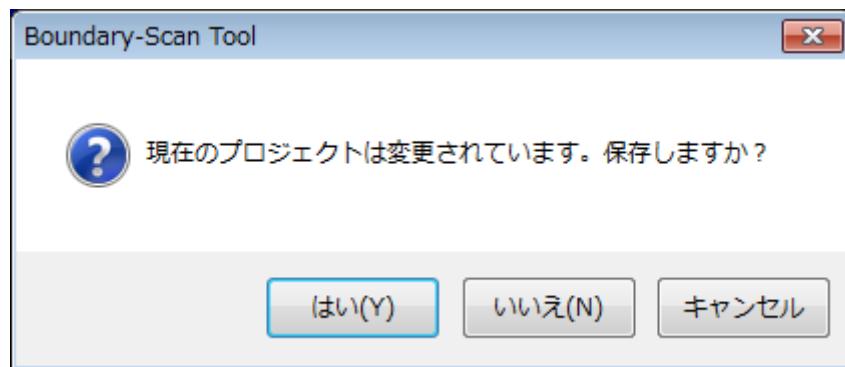
3.1.4. プロジェクトを閉じる

現在開いているプロジェクトを閉じます。

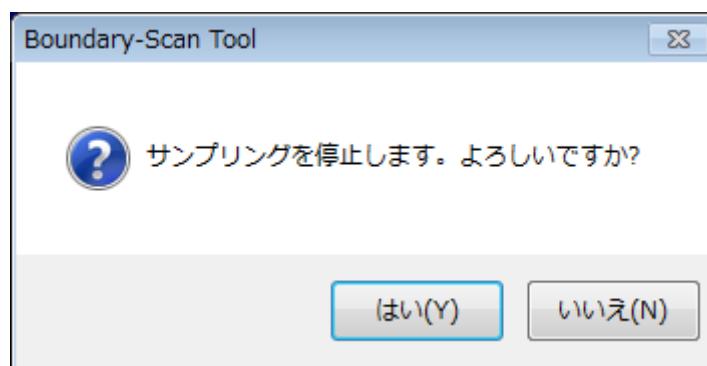
ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → プロジェクトを閉じる	---

上記メニューを実行すると、現在開いているプロジェクトが閉じプロジェクトビューから表示が消えます。このとき、プローブと接続状態であれば切断します。

プロジェクトが既に作成されていて変更があった状態（プロジェクト名の右に*が付いている）のときに新規作成・開く・閉じるメニューを実行すると、現在開いているプロジェクトを保存するかどうか確認のメッセージが表示されます。「はい」「いいえ」「キャンセル」から目的の動作を選択して先にお進みください。



プロジェクトの新規作成・開く・閉じるメニューの実行時に波形表示ウィンドウがサンプリング中だった場合、**計測を停止するかどうかの確認**と**波形を保存するかどうかの確認**メッセージが表示されます。「はい」「いいえ」「キャンセル」から目的の動作を選択して先にお進みください。



3.1.5. 波形ウインドウの新規作成

空の波形表示パターンを作成して波形表示ウインドウに表示します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 波形ウインドウの新規作成	Ctrl + W

3.1.6. 波形の保存

観測した波形データを、波形データファイルに保存します。

ファイルの拡張子は .bsw となります。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 波形の保存	---



保存したファイルはメニューの **ファイル → 波形をビューアで開く** で読み込むことができます。

3.1.7. 波形をビューアで開く

保存された波形データファイルを専用ビューアで読み込みます。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 波形をビューアで開く	---

上記メニューを実行すると、ファイル選択ダイアログが表示され、保存してある波形ファイル（拡張子は.bsw）を選択すると波形ウインドウでその波形を開きます。波形を開いた場合、以下の機能は使用できなくなります。

- ポートの追加
- グループの作成
- ポートグループの削除
- サンプリング開始
- サンプリング停止
- サンプリング間隔指定

3.1.8. 端子状態一覧の新規作成/開く

空の端子状態一覧リストを作成して端子状態一覧ウインドウに表示します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 端子状態一覧の新規作成/開く	Ctrl + L

3.1.9. 端子状態一覧を CSV 形式で保存

現在の端子状態一覧リストのデータを、CSV ファイルに保存します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 端子状態一覧を CSV ファイルに保存	---

3.1.10. 接続テストウィンドウの新規作成

空の接続テストを作成して接続テストウィンドウに表示します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 接続テストウィンドウ新規作成/開く	Ctrl + T

3.1.11. ネットリストを開く

専用のネットリストファイルを開いて接続テストウィンドウに接続設定を表示します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → ネットリストを開く	Ctrl+Shift+T

3.1.12. ネットリストの保存

接続テストウィンドウ上の接続設定を専用のネットリストファイルに保存します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → ネットリストの保存	---

3.1.13. 接続テスト結果の保存

接続テストウィンドウでのテスト結果を CSV ファイルに保存します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 接続テスト結果の保存	---

3.1.14. 印刷プレビュー

印刷プレビューを表示します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → 印刷プレビュー	---

現在表示している内容を 1 ページに表示します。サンプリング間隔、印刷している範囲、1 目盛あたりの時間、カーソルの時間、マーカの時間を下に表示します。

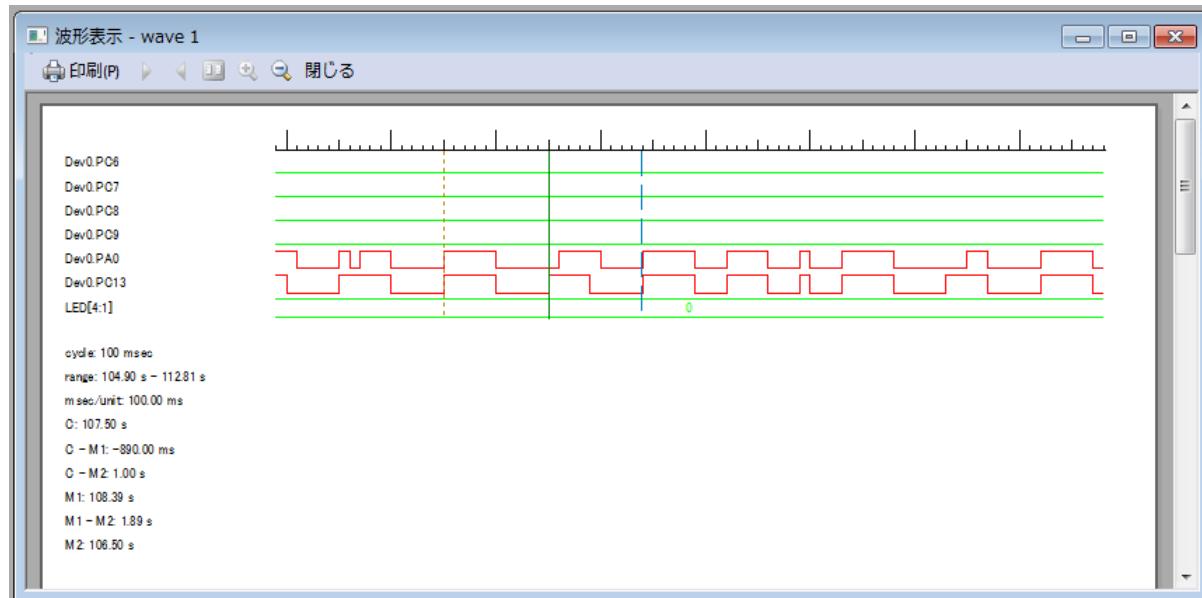


Figure 64

3.1.15. プリンターの設定

使用するプリンターに関する設定を変更します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → プリンターの設定	---

3.1.16. 印刷

観測した波形を印刷します。現在表示している内容を 1 ページに印刷します。サンプリング間隔、カーソルの時間、マーカの時間、1 目盛あたりの時間を下に印刷します。

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → 印刷	Ctrl + P

以下のダイアログが表示されるので、プリンター、印刷範囲、部数等を指定します。

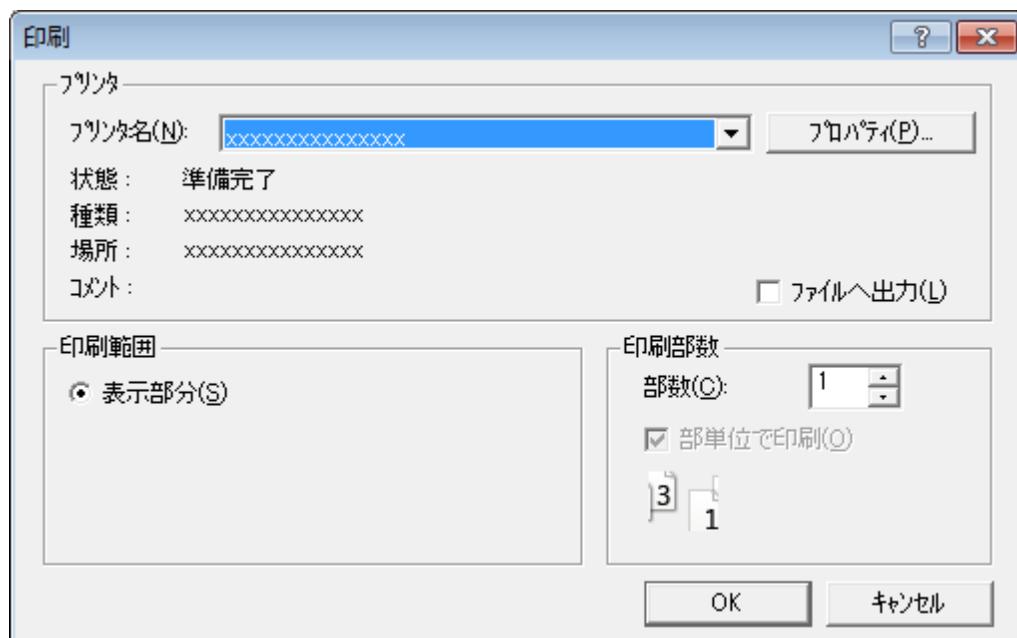
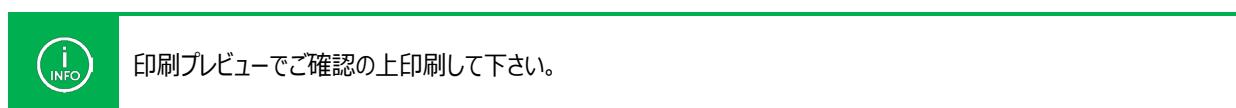


Figure 65



3.1.17. アプリケーションの終了

Boundary-Scan Tool を終了します。

ツールボタン	メニューバーの操作	ショートカットキー
---	ファイル → アプリケーションの終了	---

3.2. 表示

「表示」メニューでは、ウィンドウの表示/非表示、ツールバーのカスタマイズを行います。

3.2.1. ツールバーとドッキングウィンドウ

各ツールバーの表示・非表示の設定、項目のカスタマイズを行います。

1) 標準

標準ツールバーの表示・非表示を指定します。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → ツールバーとドッキングウィンドウ → 標準	---

2) スクリプト

スクリプトペインの表示・非表示を指定します。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → ツールバーとドッキングウィンドウ → スクリプト	---

3) プロジェクトビュー

プロジェクトビューツールバーの表示・非表示を指定します。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → ツールバーとドッキングウィンドウ → プロジェクトビュー	---

4) カスタマイズ

ツールバーのカスタマイズを行います。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → ツールバーとドッキングウィンドウ → カスタマイズ	---

3.2.2. ステータスバー

ステータスバーの表示・非表示を指定します。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → ステータスバー	---

3.2.3. スクリプト

スクリプトペインの表示・非表示を指定します。

ツールボタン	メニューバーの操作	ショートカットキー
---	表示 → スクリプト	---

3.3. プロジェクト

「プロジェクト」メニューでは、デバイス、プローブとの通信設定、スクリプトファイル、ポートグループの追加を行います。

3.3.1. 接続

本ソフトウェアとプローブを接続します。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → 接続	Ctrl + C

3.3.2. 切断

本ソフトウェアとプローブとの接続を切ります。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → 切断	Ctrl+Shift+C

3.3.3. デバイスの追加

プロジェクトビューの"ボード"ツリーにデバイスを追加します。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → デバイスの追加	---

3.3.4. 整合性テスト

バウンダリスキャンテストが実行可能かどうかをテストします。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → 整合性テスト	---

3.3.5. スクリプトファイルを追加

プロジェクトビューの"スクリプト"ツリーに Python スクリプトのファイルを追加します。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → スクリプトファイルを追加	---

3.3.6. ポートグループを追加

プロジェクトビューの"ポートグループ"ツリーにグループを追加します。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → ポートグループを追加	Ctrl + G

3.3.7. JTAG クロックの設定

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → JTAG クロックの設定	---

バウンダリスキャンテストを行う際の周波数を設定します。上記メニューを選択すると、"JTAG クロックの設定"ダイアログが表示されます。プルダウンリストの中から希望する周波数を選択するか（1MHz 以上のクロック）、プルダウンリストで"低速クロック設定"を選択した上で"低速クロックの周波数"エディットボックスに周波数を入力してください（625kHz 未満のクロック）。周波数を指定した後、OK をクリックした時点で選択した周波数に設定されます。

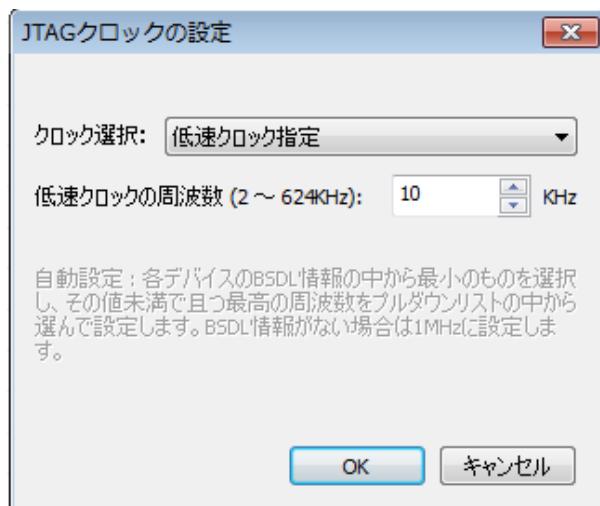


Figure 66



OK 後は、指定した周波数で動作テストを行い、テストに通ればその周波数になります。通らなかつた場合はテストが通るまで 1 段階ずつ周波数を下げていきます。"自動設定"を選択したときは、各デバイスの BSDL 情報に記述されている動作周波数の中で最小のものを選択し、その値未満で且つ最高の周波数をプルダウンリスト中から選んで設定します。

3.3.8. JTAG 信号の強制出力

デバイスの自動認識時やバウンダリスキャンテスト実行時に、ターゲットの電源状態を確認せずに JTAG 信号を出力するかどうかを設定します。

電源リファレンス信号が結線されておらず、電源状態を検出できないボードのバウンダリスキャンテストを行う際に チェックを付けてください。

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → JTAG 信号の強制出力	---

3.4. 波形表示

「波形表示」メニューでは、バウンダリスキャンの開始/停止をコントロールします。

3.4.1. サンプリング開始（計測開始）

計測を開始します。

ツールボタン	メニューバーの操作	ショートカットキー
	波形表示 → サンプリング開始	F7

3.4.2. サンプリング停止（計測停止）

計測を停止します。

ツールボタン	メニューバーの操作	ショートカットキー
	波形表示 → サンプリング停止	Ctrl + F7

3.5. 端子状態一覧

「端子状態一覧」メニューでは、端子状態一覧ウインドウ関連のコントロールを行います。

3.5.1. SAMPLE コマンド

バウンダリスキャンチェーン上の各デバイスに SAMPLE 命令を与えてポートの状態を取得します。

ツールボタン	メニューバーの操作	ショートカットキー
	端子状態一覧 → SAMPLE コマンド	F5

3.5.2. EXTEST コマンド実行

バウンダリスキャンチェーン上の各デバイスに EXTEST 命令を与えて各デバイスの入出力ピンの接続状態を確認します。

ツールボタン	メニューバーの操作	ショートカットキー
	端子状態一覧 → EXTEST コマンド	Ctrl + F5

3.5.3. EXTEST 実行時に警告を表示

このメニューにチェックが付いているときは、EXTEST コマンドメニューの実行時に"この設定で EXTEST を実行してよいか"の旨のメッセージが表示されます。

3.6. 接続テスト

「接続テスト」メニューでは、接続テストウィンドウ関連のコントロールを行います。

3.6.1. 期待値の学習

接続テストウィンドウの各ネットの出力値に応じた期待値を作成します。

ツールボタン	メニューバーの操作	ショートカットキー
	接続テスト → 期待値の学習	---

3.6.2. 接続テスト

接続テストウィンドウの各ネットの接続テストを行います。

ツールボタン	メニューバーの操作	ショートカットキー
	接続テスト → 接続テスト	---

3.6.3. 信号出力時に警告を表示

このメニューにチェックが付いているときは、期待値の学習メニューと接続テストメニューの実行時に"この設定で EXTEST を実行してよいか"の旨のメッセージが表示されます。

3.7. ツール

「ツール」メニューでは、BSDL、ライセンス、エディッタの設定を行います。

3.7.1. BSDL 管理

ツールボタン	メニューバーの操作	ショートカットキー
---	ツール → BSDL 管理	Ctrl + B

本ソフトウェアで使用する BSDL ファイルの管理を行います。

上記メニューを選択すると、"BSDL"管理ダイアログが表示されます。メーカーリストでメーカーを選択すると、本ソフトウェアに登録されているそのメーカーのデバイスがリストに一覧表示されます。新たにデバイスを追加する場合は、"BSDL エディットボックス"に BSDL ファイルのパスを直接入力するか、 [...]ボタンをクリックしてファイル選択を行ってください。[Read]ボタンをクリックすることで指定した BSDL ファイルが読み込まれ、デバイスが一覧に追加されます。

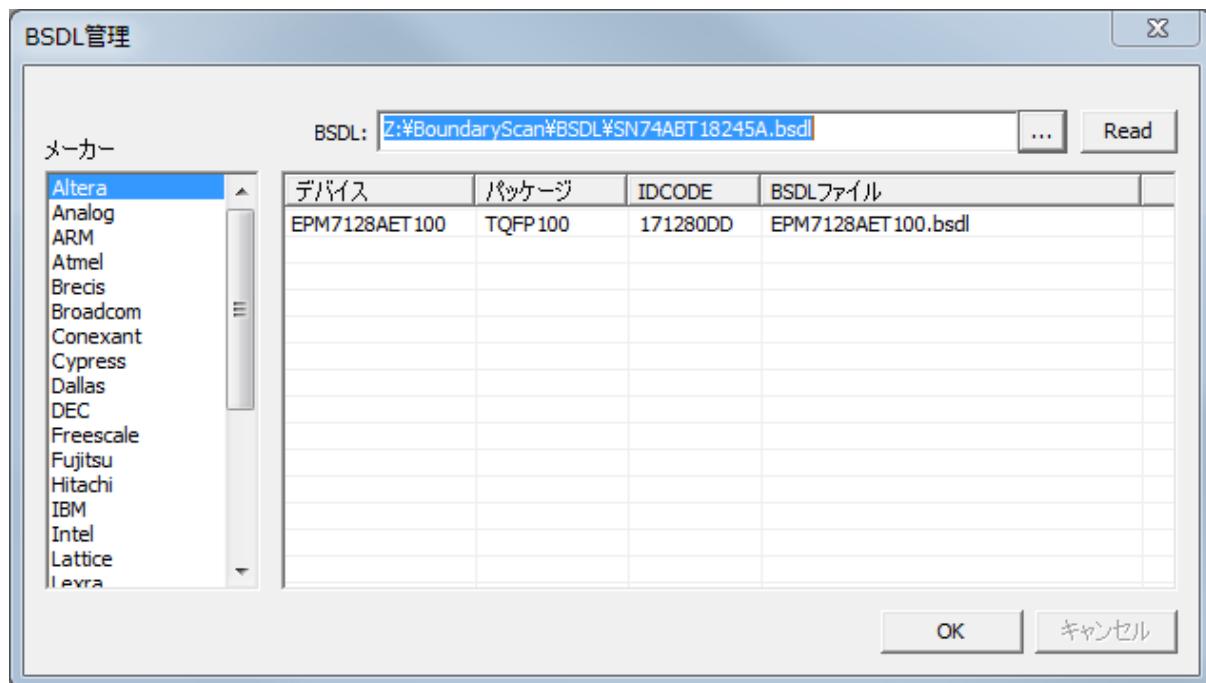


Figure 67



既に登録済みのデバイスの BSDL ファイルを読み込ませた場合は、読み込ませた情報で上書きされます。

3.7.2. プローブのライセンス取得

ツールボタン	メニューバーの操作	ショートカットキー
---	ツール → プローブのライセンス取得	---

本ソフトウェアはライセンスシステムを採用しています。
プローブの Serial No.を弊社にご連絡いただくと、License Code を発行します。

①License Code の入力について

プローブ接続時、[プローブ選択]ダイアログに「ライセンス未登録」または、「接続可能」と表示されているものを選択し、
[ライセンス登録] ボタンをクリックすると、License Code の入力ウィンドウが表示されます。

弊社からお送りした License Code を入力していただくと、対象のプローブで本ソフトウェアを使用することができるようになります。



Figure 68

②License Code 入力時の注意事項

License Code の入力は、お送りした情報通りに入力してください。



License Code を送付したメールの内容と一致しているにもかかわらず受け付けられない場合は、当社までお問い合わせください。

③エラー表示一覧

Table 13

表示	意味	対処方法
License Code が不正です。	License Code の書式が不正など、 License Code の解析ができない場合 に表示されます。	License Code が送られてきたメールを ご確認の上、再度 License Code を入 力してください。
License Code と Serial No が一致 しません。	ご使用のプローブの Serial No と、 License Code が対象としているプロー ブの Serial No が一致していない場合に 表示されます。	License Code が送られてきたメールを ご確認の上、再度 License Code を入 力してください。
License Code と 本体の Serial No の組み合わせをご確認ください。		

3.7.3. エディタ指定

ツールボタン	メニューバーの操作	ショートカットキー
---	ツール → エディタ指定	---

プロジェクトビューのスクリプトファイルのコンテキストメニューから起動するエディタを指定します。

上記メニューを選択すると、"エディタ指定"ダイアログが表示されます。ダイアログには現在指定されているエディタが表示されます。初期値は『メモ帳』(notepad.exe) です。エディットボックスにエディタの実行ファイルを直接入力するか、ファイル選択ボタン(『...』)をクリックしてファイル選択ダイアログから 指定してください。
エディタはフルパスで指定してください。

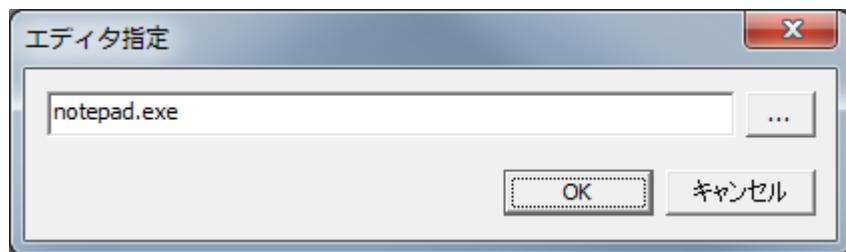


Figure 69

3.8. ヘルプ[°]

「ヘルプ」メニューでは、ヘルプ、バージョン情報を表示します。

3.8.1. ヘルプの表示

本ソフトウェアの取扱説明書を表示します。

当社 Web 上の取り扱い説明書 URL が開きます。

ツールボタン	メニューバーの操作	ショートカットキー
---	ヘルプ → ヘルプの表示	F1

3.8.2. バージョン情報

本ソフトウェアのバージョンを表示します。

ツールボタン	メニューバーの操作	ショートカットキー
	ヘルプ → バージョン情報	---

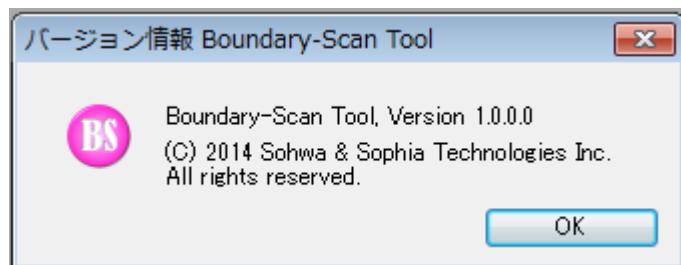


Figure 70

4. ウィンドウ

各種ウィンドウの説明をします。

4.1. プロジェクトビューウィンドウ

プロジェクトの各種設定を表示するウィンドウです。

各表示項目について説明します。

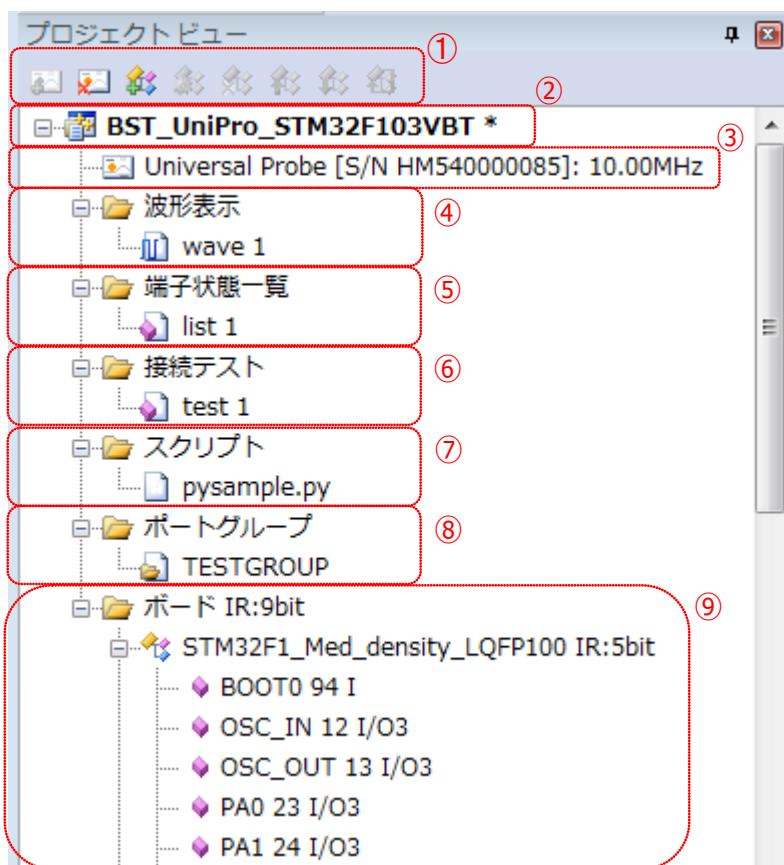


Figure 71

- ① [ツールバー](#)
- ② [プロジェクト名](#)
- ③ [ブロープ情報](#)
- ④ [波形表示ツリー](#)
- ⑤ [端子状態一覧ツリー](#)
- ⑥ [接続テストツリー](#)
- ⑦ [スクリプトツリー](#)
- ⑧ [ポートグループツリー](#)
- ⑨ [ボードツリー](#)

4.1.1. ツールバー

プロジェクトビューのメニューのうち、代表的なものをツールバーから使用できるようにしています。

4.1.2. プロジェクト名

現在使用中のプロジェクト名が表示されます。プロジェクトファイル名から拡張子を除いた文字列がプロジェクト名となります。

プロジェクトに変更があると、プロジェクト名の右側に"*"が表示されるようになります。

この"*"はプロジェクトの保存を行うことで表示されなくなります。

以下のときにプロジェクトは変更があったと判断されます。

- 接続するプローブが決まったとき・変更されたとき
- プローブの JTAG クロックが変更されたとき
- 波形表示パターンが新規作成されたとき・名前が変更されたとき・削除されたとき
- 波形表示パターンに登録されたポートの数が変わったとき
- 波形表示パターンに登録されているポートの順番が変わったとき
- 端子状態一覧リストが新規作成されたとき・名前が変更されたとき・削除されたとき
- 端子状態一覧リストに登録されたポートの数が変わったとき
- 端子状態一覧リストに登録されているポートの順番が変わったとき
- 端子状態一覧リストの各ポートのチェック状態が変わったとき
- "ボード"タグのデバイス構成が変わったとき
- "JTAG 信号の強制出力"メニューのチェック状態が変わったとき

4.1.3. プローブ情報

接続するプローブの情報を表示します。

プローブとの接続状態によって表示が以下のように変化します。

Table 14

初期値	Unknown
プローブ接続時	UNIVERSAL PROBE [プローブシリアル No] : JTAG クロック
プローブ切断時	UNIVERSAL PROBE [プローブシリアル No] : JTAG クロック: 未接続

また、コンテキストメニューは以下の通りです。

Table 15

接続	プローブと接続します。
切断	プローブとの接続を切れます。

4.1.4. 波形表示ツリー

"波形表示パターン"の管理を行います。

"波形表示"欄のコンテキストメニューは以下の通りです。

Table 16

波形表示	"波形表示パターン"を新規作成します。
------	---------------------

新規作成された"波形表示パターン"のデフォルト名は"wave n"（n は 1 からの通し番号）となります。
 "wave n"のコンテキストメニューは以下の通りです。

Table 17

波形表示	選択された"wave n"を波形表示ウィンドウで表示します。
名前変更	選択された"wave n"を任意の名称に変更します。
削除	選択された"wave n"をプロジェクトから削除します。

4.1.5. 端子状態一覧ツリー

"端子状態一覧リスト"の管理を行います。

"端子状態一覧"欄のコンテキストメニューは以下の通りです。

Table 18

端子状態一覧	"端子状態一覧リスト"を新規作成します。
--------	----------------------

新規作成された"端子状態一覧リスト"のデフォルト名は"list n"（n は 1 からの通し番号）となります。
 "list n"のコンテキストメニューは以下の通りです。

Table 19

端子状態一覧	選択された"list n"を端子状態一覧ウィンドウで表示します。
名前変更	選択された"list n"を任意の名称に変更します。
削除	選択された"list n"をプロジェクトから削除します。

4.1.6. 接続テストツリー

"接続テストリスト"の管理を行います。

"接続テスト"欄のコンテキストメニューは以下の通りです。

Table 20

接続テスト	"接続テストリスト"を新規作成します。
ネットリストを開く	専用のネットリストファイルを開いて接続テストを構成します。

新規作成された"接続テストリスト"のデフォルト名は"test n"（n は 1 からの通し番号）となります。
 "test n"のコンテキストメニューは以下の通りです。

Table 21

接続テスト	選択された"test n"を接続テストウィンドウで表示します。
名前変更	選択された"test n"を任意の名称に変更します。
削除	選択された"test n"をプロジェクトから削除します。

4.1.7. スクリプトツリー

"スクリプトファイル"の管理を行います。

"スクリプト"欄のコンテキストメニューは以下の通りです。

Table 22

ファイルを追加	"スクリプトファイル"を追加します。
---------	--------------------

追加された Python スクリプトファイル名が"スクリプト"ツリーの下に表示されます。

"スクリプトファイル"のコンテキストメニューは以下の通りです。

Table 23

実行	選択されたスクリプトファイルを実行します。結果等はスクリプトペインで確認できます。
削除	選択されたスクリプトファイルをプロジェクトから削除します。実際のファイルは削除しません。
エディタ起動	選択されたスクリプトファイルをエディタで開きます。エディタの指定はツールメニューから行います。

4.1.8. ポートグループツリー

任意のポートを一つにまとめた"グループ"の管理を行います。

"ポートグループ"欄のコンテキストメニューは以下の通りです。

Table 24

グループの追加	グループ作成ダイアログが開きます。"グループ"を新規作成します。
---------	----------------------------------

グループ作成ダイアログで設定されたグループ名が"ポートグループ"ツリーの下に表示されます。

"グループ"のコンテキストメニューは以下の通りです。

Table 25

グループの編集	選択された"グループ"を編集します。グループ作成ダイアログが開くので、変更したい箇所を編集します。
削除	選択された"グループ"をプロジェクトから削除します。

4.1.9. ボードツリー

"ボード"欄には"デバイスの自動認識"や"整合性テスト"を行ったときにターゲットボードから取得した 全デバイスの IR レジスタの総ビット長が表示されます。

初期値は"0"で、"接続"メニュー実行時や"整合性テスト"メニュー実行時に更新されます。

また、"デバイス"の管理を行います。

デバイスは情報の充実具合によって以下のように表示が変わります。

Table 26

追加直後	Unknown BSDL 情報がない状態です。
デバイス編集直後	デバイス名 IR:XXXX "デバイス編集"でデバイスを指定しただけの状態です。IR レジスタ長が求まっていません。
IR レジスタ長編集後	デバイス名 IR:n bit 整合性テストに通過した・デバイスの自動認識に成功した IR レジスタ長を手動で変更したときにこの表示になります。

"ボード"欄のコンテキストメニューは以下の通りです。

Table 27

デバイスの追加	"ボード"ツリーにデバイスを新規作成します。
整合性テスト	プロジェクトに登録したデバイスでバウンダリスキャンテストを実行できるかテストします。

新規作成された"デバイス"のデフォルト名は"Unknown"となります。

"デバイス"のコンテキストメニューは以下の通りです。

Table 28

デバイス編集	"デバイス編集"ダイアログを開きます。選択したデバイスの情報を更新します。
デバイス削除	選択したデバイスをプロジェクトから削除します。
デバイスの上移動	選択したデバイスを一つ上のデバイスに入れ替えます。プロジェクト上でのバウンダリスキャンチェーンの位置も入れ替わります。
デバイスの下移動	選択したデバイスを一つ下のデバイスに入れ替えます。プロジェクト上でのバウンダリスキャンチェーンの位置も入れ替わります。
デバイスプロパティ	選択したデバイスの情報を表示します。"デバイスプロパティ"ダイアログでは、IR レジスタのビット長の変更とバイパス設定ができます。

各デバイスの下には BSDL 情報から求めた"ポート"が表示されます。デバイス、またはポートは波形表示ウィンドウや端子状態一覧ウィンドウにドラッグ＆ドロップすることで、該当する"波形表示パターン"及び"端子状態一覧リスト"に登録することができます。各ポートは、"ポート名" "ピン番号" "タイプ"の並びで表示されます。

4.2. 波形表示ウィンドウ

計測した波形を表示するウィンドウです。1 目盛あたりの時間はステータスバーの最後に表示されます。また波形は入力のポートは赤、出力のポートは緑、ハイインピーダンスのときは青で表示されます。

各表示項目について説明します。

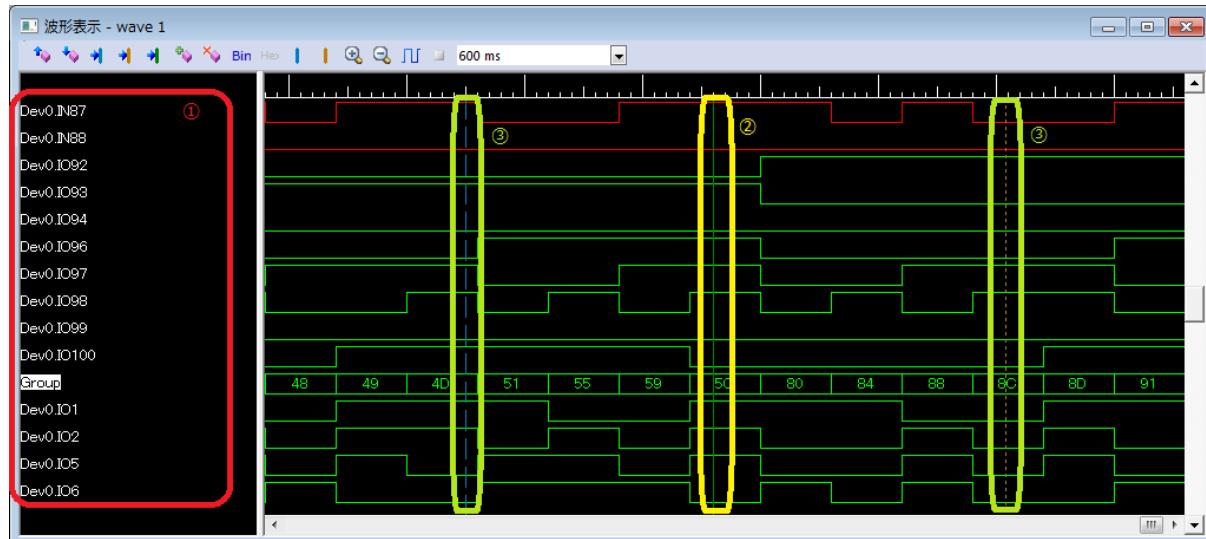


Figure 72

- ① ポートグループ名表示欄
- ② カーソル（緑の実線）
- ③ マーカ（青・橙の破線）

4.2.1. ポートグループ名表示欄

ポートグループ名が表示されます。

ポートは "Dev" + JTAG チェーン番号 + "." + ポート名で表示されます。

グループ名の場合、その上にマウスカーソルを合わせることにより、バスを構成しているポートを MSB からツールチップで表示します。

ポートグループを選択し、マウスを右クリックすることにより表示されるコンテキストメニューから、以下の操作を行うことができます。

- 移動
- 削除

4.2.2. カーソル（緑の実線）

カーソルの位置を緑色の線で示します。

4.2.3. マーカ（青・橙の破線）

マーカの位置を示します。

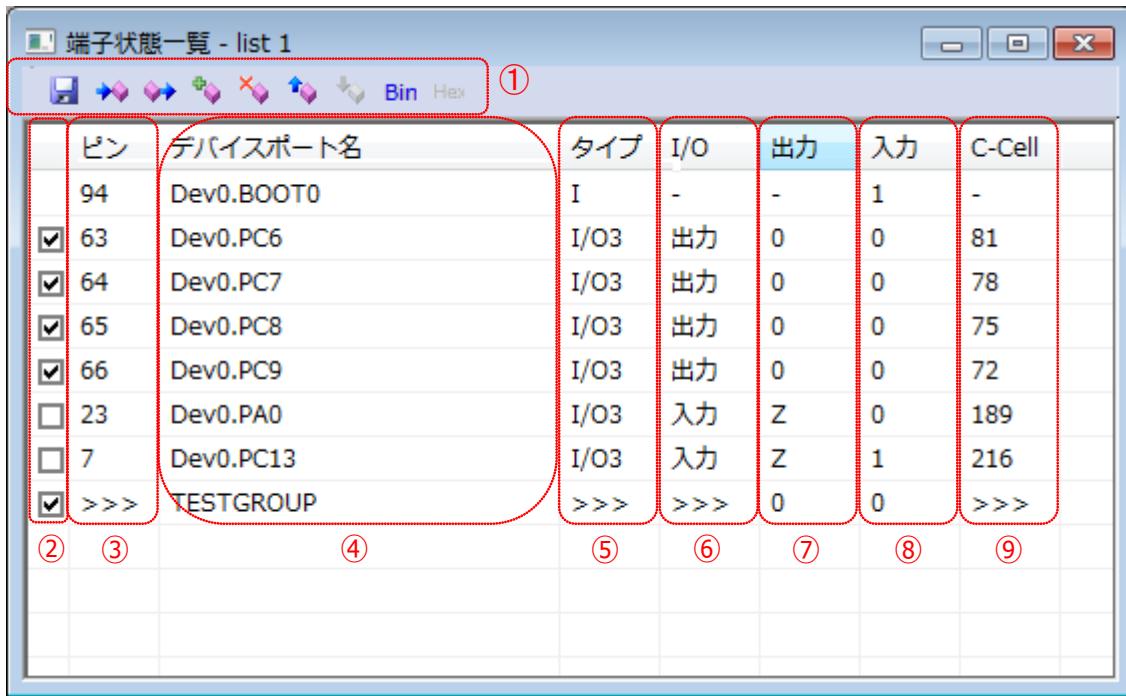
マーカは 2箇所設定することができます。

- マーカ 1：青の破線で表示します。
- マーカ 2：オレンジの破線で表示します。

4.3. 端子状態一覧ウィンドウ

各デバイスの端子状態の監視とバウンダリスキャンテストを行うためのウィンドウです。

各表示項目について説明します。



Pin	Device Port Name	Type	I/O	Output	Input	C-Cell
94	Dev0.BOOT0	I	-	-	1	-
<input checked="" type="checkbox"/> 63	Dev0.PC6	I/O3	出力	0	0	81
<input checked="" type="checkbox"/> 64	Dev0.PC7	I/O3	出力	0	0	78
<input checked="" type="checkbox"/> 65	Dev0.PC8	I/O3	出力	0	0	75
<input checked="" type="checkbox"/> 66	Dev0.PC9	I/O3	出力	0	0	72
<input type="checkbox"/> 23	Dev0.PA0	I/O3	入力	Z	0	189
<input type="checkbox"/> 7	Dev0.PC13	I/O3	入力	Z	1	216
<input checked="" type="checkbox"/> >>>	TESTGROUP	>>>	>>>	0	0	>>>

Figure 73

- ① ツールバー
- ② チェックボックスカラム
- ③ ピンカラム
- ④ デバイスポート名カラム
- ⑤ タイプカラム
- ⑥ I/O カラム
- ⑦ 出力カラム
- ⑧ 入力カラム
- ⑨ C-Cell (コントロールセル) カラム

4.3.1. ツールバー

端子状態一覧ウィンドウのメニューのうち、代表的なものをツールバーから使用できるようにしています。

4.3.2. チェックボックスカラム

出力専用ポート、もしくは入出力ポートはここにチェックボックスが表示されます。入力専用ポートには表示されません。

グループは、構成ポートのどれか一つに出力専用ポートか入出力ポートがあればチェックボックスが表示されます。全て入力専用ポートであれば表示されません。チェックを付けたポートは以下の項目が設定可能になります。

Table 29

出力専用ポート	"出力"カラムをクリックすることで、出力値を設定することができます。
入出力ポート	"I/O"カラムをクリックすることで、ポートの入出力方向を設定することができます。
グループ	"出力"カラムを"出力"にした上で"出力"カラムをクリックすることで、出力値を設定することができます。

チェックを付けなかったポートは、バウンダリスキャンテスト時に以下のように扱われます。

Table 30

出力専用ポート	出力値"Z"（Disable）とします。（コントロールセルを持つポートの場合） 出力値"0"（Low）とします。（コントロールセルを持たないポートの場合）
入出力ポート	入力ポートとします。



このカラムのチェックを変更したとき、このポートの他に C-Cell カラムの番号が同じポートがあれば、そのポートのチェックも併せて変更されます。
グループのチェックを変更したとき、構成する全ポートのチェック状態が切り替わります。

4.3.3. ピンカラム

そのポートの、デバイス上のピン番号を表示します。

グループの場合は『>>>』と表示されます。当該カラムにマウスカーソルを合わせるとグループの構成ポートのピン番号がポップアップ表示されます。

4.3.4. デバイスポート名カラム

ポート名を表示します。デフォルトの表示形式は"Dev n.ポート名"です。

(n は 0 から始まるバウンダリスキャンチェーン上のデバイス位置)

"デバイス参照名の設定"や"ポート名の変更"で名前を変更した場合は、その名前で表示されます。

グループの場合はグループ名を表示します。当該カラムにマウスカーソルを合わせるとグループの構成ポート名がポップアップ表示されます。

4.3.5. タイプカラム

そのポートの入出力タイプを表示します。以下の値が表示されます。

Table 31

I	入力専用ポートです。
O	出力専用ポートです。
O3	出力専用ポートです。BSDL で "function" が "OUTPUT3" と定義されている、3 ステート出力を持つポートです。
I/O	入出力ポートです。
I/O3	入出力ポートです。BSDL で "function" が "OUTPUT3" と定義されている、3 ステート出力を持つポートです。
>>>	グループです。当該カラムにマウスカーソルを合わせるとグループの構成ポートのタイプがポップアップ表示されます。

4.3.6. I/O カラム

ポートの入出力方向を表示/設定します。

入出力ポートのチェックカラムにチェックが付いているときに、"入力"か"出力"を選択することができます。

入力専用ポート及び出力専用ポートでは、このカラムの値は"-”固定となります。

グループは『>>>』が表示されます。値の選択はできません。

当該カラムにマウスカーソルを合わせるとグループの構成ポートの I/O がポップアップ表示されます。

"入力"から"出力"、"出力"から"入力"に値を変更したとき、"出力"カラムの値も以下のように変更されます。

Table 32

"入力" > "出力"	"出力"カラムは"0"（Low 出力）になります。
"出力" > "入力"	"出力"カラムは"Z"（ハイインピーダンス）になります。



このカラムの値を変更したとき、このポートの他に C-Cell カラムの番号が同じポートがあれば、そのポートの値も併せて変更されます。

4.3.7. 出力カラム

ポートの出力値を表示/設定します。

出力専用ポート及び出入力ポートのチェックカラムにチェックが付いているときに出力値を選択することができます。

入力専用ポートでは、このカラムの値は"--"固定となります。

出力値として選択できる値は以下の通りです。

Table 33

0	Low 出力です。"O"・"O3"・"I/O"・"I/O3"のポートで選択できます。
1	High 出力です。"O"・"O3"・"I/O"・"I/O3"のポートで選択できます。
Z	ハイインピーダンス出力です。"O3"のポートで選択できます。



グループでは、このカラムの値はエディットボックスで設定します。グループの表示フォーマットに従って 16 進数か 2 進数で値を設定します。その際、以下に該当するビット位置は値が無視されます。

- ・入力専用ポートのビット位置
- ・I/O カラムが"入力"の出入力ポートのビット位置
- ・個別にチェックを外した出力ポートや出入力ポートのビット位置

4.3.8. 入力カラム

ポートの入力値を表示します。

出力専用ポートでは、このカラムの値は"--"固定となります。

4.3.9. C-Cell（コントロールセル）カラム

そのポートのコントロールセル番号が BSDL に定義されていればその番号が表示されます。

定義されていないポートでは、このカラムの値は"--"となります。

グループの場合は『>>>』が表示されます。当該カラムにマウスカーソルを合わせるとグループの構成ポートのコントロールセル番号がポップアップ表示されます。

4.4. 接続テストウィンドウ

バウンダリスキャンテストを用いて各デバイスの信号間の接続をテストするためのウィンドウです。
各表示項目について説明します。

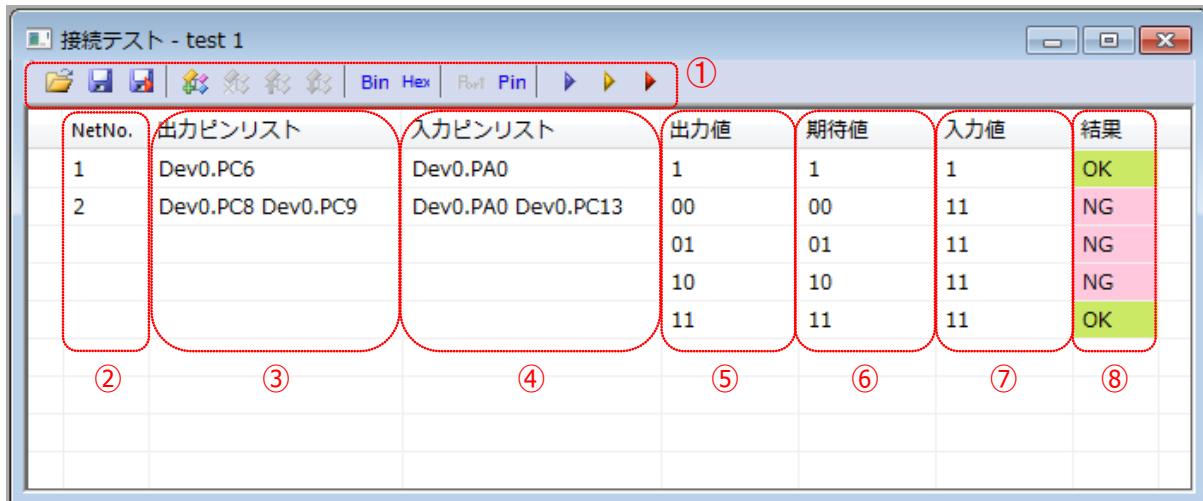


Figure 74

- ① [ツールバー](#)
- ② [Net 番号カラム](#)
- ③ [出力ピンリストカラム](#)
- ④ [入力ピンリストカラム](#)
- ⑤ [出力値カラム](#)
- ⑥ [期待値カラム](#)
- ⑦ [入力値カラム](#)
- ⑧ [結果カラム](#)

4.4.1. ツールバー

接続テストウィンドウメニューのうち、主要なものをツールバーから使用できるようにしています。

4.4.2. Net 番号カラム

端子間の接続設定ごとに振られる通し番号です。一番上のネットが常に番号 1 となります。ネットの追加、削除、上下移動を行うと番号を振り直します。

4.4.3. 出力ピンリストカラム

接続テスト実行時に信号を出力する側のピン名やグループ名の一覧です。

入力専用端子を登録すると設定不備としてカラムの背景色がピンク色になります。

リストの先頭（一番左）のピンが出力値カラムの MSB、というように出力値カラムと一対一対応しています。

当該カラムにマウスカーソルを合わせるとグループの構成ポートのピン番号がポップアップ表示されます。

4.4.4. 入力ピンリストカラム

接続テスト実行時に信号が入力される側のピン名やグループ名の一覧です。

出力専用端子を登録すると設定不備としてカラムの背景色がピンク色になります。

リストの先頭（一番左）のピンが期待値カラム及び入力値カラムの MSB、というように期待値や入力値と一対一対応しています。

当該カラムにマウスカーソルを合わせるとグループの構成ポート名がポップアップ表示されます。

4.4.5. 出力値カラム

接続テスト実行時に出力ピンリストの各端子から出力する値を設定します。

出力値カラムの MSB が出力ピンリストの先頭（一番左）、というように出力ピンリストカラムと一対一対応しています。

16進数表示されているときは値の末尾に文字"H"がつきます。

出力ピンリストカラムを編集すると未設定の状態になります。

4.4.6. 期待値カラム

接続テスト実行時に入力ピンリストの各端子が取るはずの値を設定します。

期待値カラムの MSB が入力ピンリストの先頭（一番左）、というように入力ピンリストカラムと一対一対応しています。

文字"x"を設定すると、該当するビットは接続テストで入力値と比較する際に無視されます（OKとして扱います）。

出力ピンリストカラム、入力ピンリストカラムおよび出力値カラムを編集すると未設定の状態になります。

4.4.7. 入力値カラム

接続テスト実行時に入力ピンリストの各端子の値を取り込んだ結果を表示します。

入力値カラムの MSB が入力ピンリストの先頭（一番左）、というように入力ピンリストカラムと一対一対応しています。

4.4.8. 結果カラム

各行の接続テスト実行時に入力値と期待値を比較した結果を表示します。

比較した結果が同じ（出力値によって、期待値通りの値が入力された）場合は"OK"を表示し、結果カラムの背景色が緑になります。

異なる（期待値通りの値が入力されなかった）場合は"NG"を表示し、結果カラムの背景色がピンク色になります。

4.5. デバイス編集/BSDL 管理ダイアログ

"デバイス編集"メニューから呼ばれる、デバイス情報を選択するためのダイアログウインドウです。
 "BSDL 管理"メニューから呼ばれる、BSDL ファイルの管理ダイアログも併せて説明します。
 各表示項目について説明します。

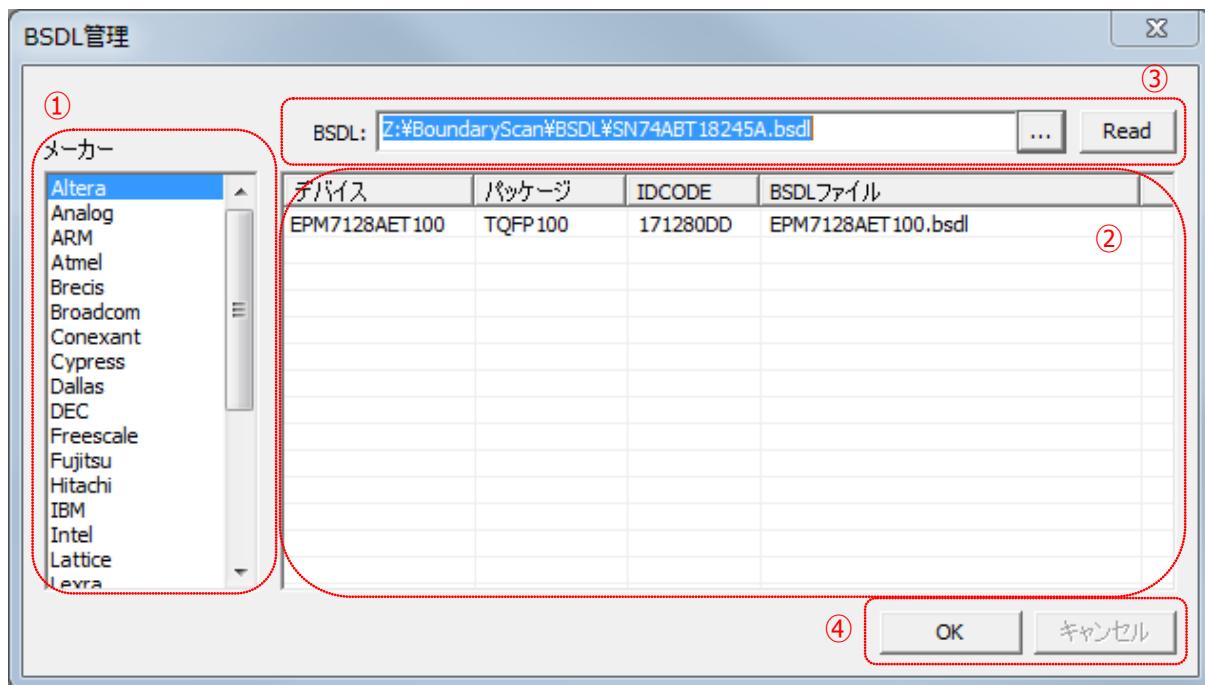


Figure 75

- ① メーカーリスト
- ② デバイスリスト
- ③ BSDL ファイルエディット
- ④ OK・キャンセル

4.5.1. メーカーリスト

デバイスマーカーを一覧表示しています。メーカーを選択するとデバイスリストにそのメーカーのデバイスが一覧表示されます。BSDL の "IDCODE" 命令などを元にしています。BSDL を解析してメーカー名が判別できなかったデバイスについては、メーカーリストの "Unknown" に登録されています。

4.5.2. デバイスリスト

メーカーを選択されたメーカーのデバイスを一覧表示しています。
 各カラムの意味は以下の通りです。

Table 34

デバイス名	BSDL から解析したデバイス名です。
パッケージ	BSDL から解析した QFP などのデバイスパッケージ情報です。
IDCODE	BSDL から解析した IDCODE 命令の値です。
BSDL ファイル	デバイス情報の解析に使用した BSDL ファイル名です。

4.5.3. BSDL ファイルエディット

"デバイスリスト"に目的のデバイスがない場合に、ここで BSDL ファイルを指定することでデバイスを追加することができます。手順は以下の通りです。

- BSDL:エディットボックスに目的のデバイスの BSDL ファイルをフルパス指定で入力します。
もしくは[...]ボタンを クリックしてファイル選択ダイアログを呼び出し、BSDL ファイルを指定します。
- [Read]ボタンを クリックして BSDL ファイルを読み込みます。
- 読み込まれた BSDL ファイルの解析に成功すると、解析したメーカーのデバイスリストにそのデバイスが登録されます。



一度読み込まれた BSDL 情報をキャンセルボタンで元に戻すことはできません。ご注意ください。



既に登録済みのデバイスの BSDL ファイルを読み込まれた場合は、読み込まれた情報で上書きされます。

4.5.4. OK・キャンセル

"デバイス編集"ダイアログと"BSDL 管理"ダイアログでの各ボタンの動作は以下の通りです。

Table 35

デバイス編集	OK	デバイスリストでデバイスが選択されたときに有効になります。 クリックするとダイアログが閉じ、プロジェクトビューのデバイス情報が選択したデバイスで更新されます。
	キャンセル	常に有効です。クリックするとダイアログが閉じます。プロジェクトには影響はありません。
BSDL 管理	OK	常に有効です。クリックするとダイアログが閉じます。プロジェクトには影響はありません。
	キャンセル	常にグレーアウトされています。

4.6. JTAG クロック設定ダイアログ

バウンダリスキャンテストを実行する際の、プローブとターゲットデバイス間のクロック周波数を設定するためのダイアログウィンドウです。各表示項目について説明します。

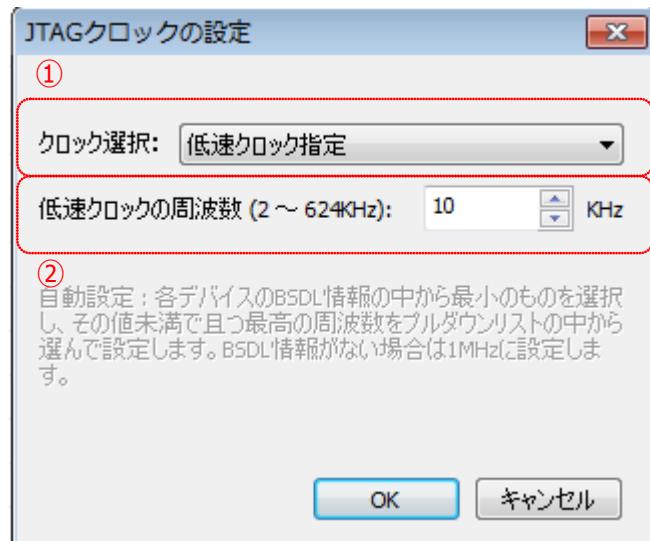


Figure 76

- ① [クロック選択](#)
- ② [低速クロックエディット](#)

4.6.1. クロック選択

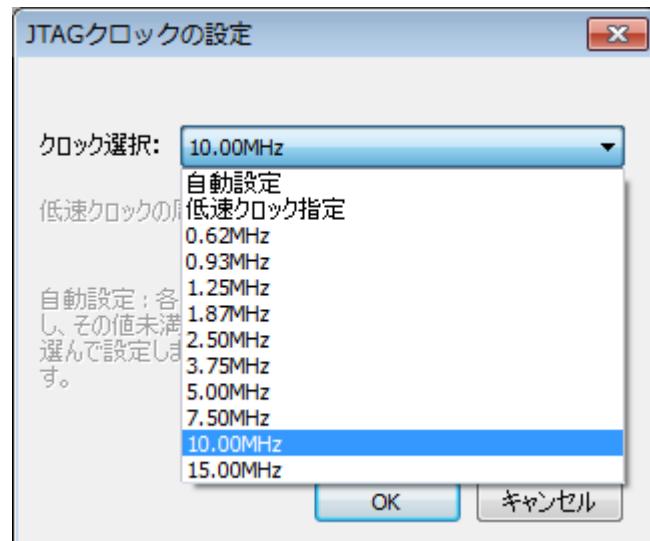


Figure 77

プローブに設定できる周波数がプルダウンリストに表示されます。

プロジェクトの初期値は"低速クロック指定"になります。

各項目の詳細は以下の通りです。

Table 36

自動設定	クロック周波数を自動で設定します。 自動認識できるターゲットに接続されているかプロジェクトにデバイスが設定されていれば、各デバイスの BSDL 情報の中から最低の周波数を選び、その値未満で且つプルダウンリストから選択できる最高の周波数を設定します。
低速クロック指定	1MHz 未満の周波数をプルダウンリスト下の 低速クロックエディットボックス で設定します。 <div style="border: 2px solid red; padding: 5px; margin-top: 10px;">  プロジェクトの初期値として、"低速クロック指定"の"10kHz"が設定されています。 </div>
その他の周波数	BSDL 情報とは無関係に動作するかしないかで値を設定するため、 BSDL に記述されている動作周波数より大きい周波数に設定される可能性があります。 <div style="border: 2px solid green; padding: 5px; margin-top: 10px;">  BSDL 情報とは無関係に動作するかしないかで値を設定するため、 BSDL に記述されている動作周波数より大きい周波数に設定される可能性があります。 </div>

4.6.2. 低速クロックエディット

1MHz 未満の、比較的低い周波数でプローブとターゲットデバイスとの JTAG 通信を行います。 [クロック選択](#) プルダウンリストで "低速クロック指定"を選択すると有効になります。 値の範囲は 2~624 (単位 : kHz) です。

4.7. スクリプトペインウィンドウ

Python スクリプトを実行するためのウィンドウです。
ファイルを指定して実行や、コマンドラインで 1 行ずつ実行することができます。
各表示項目について説明します。

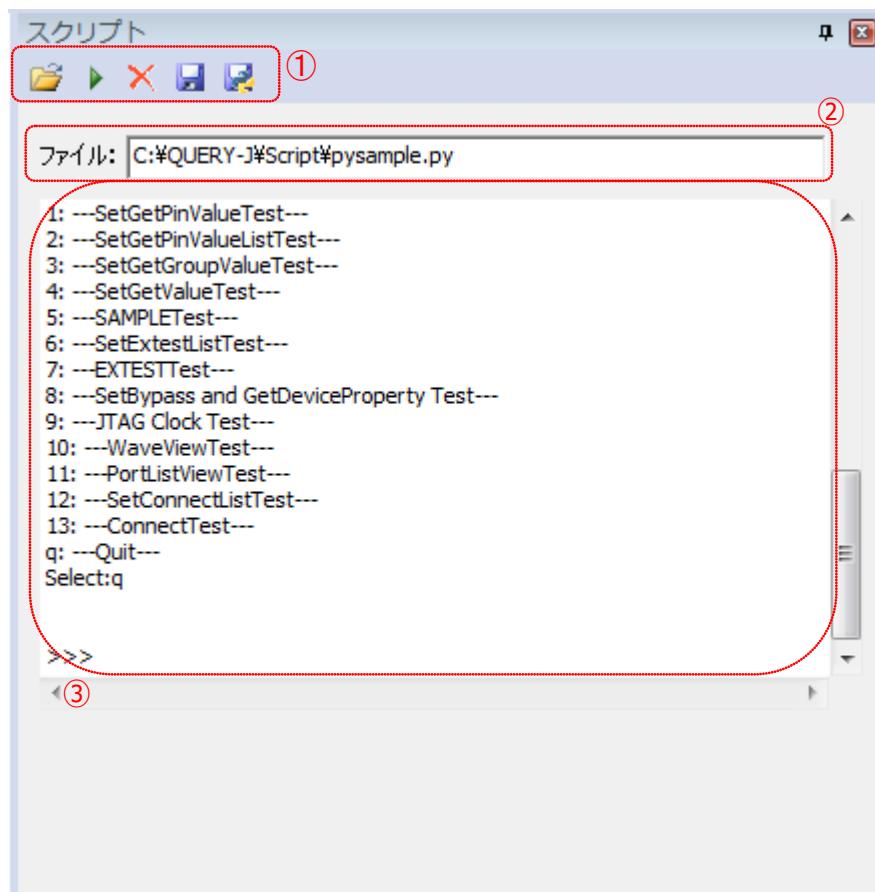


Figure 78

- ① [ツールバー](#)
- ② [ファイル名エディットボックス](#)
- ③ [標準入出力エディットボックス](#)

4.7.1. ツールバー

スクリプトペイン上で実行する機能は全てツールバーから使用します。

4.7.2. ファイル名エディットボックス

バッチファイルとして実行する Python スクリプトのファイル名を入力/表示します。
入力は以下のいずれかの方法で行います。

- ツールバーの"ファイルを開く"ボタンによりファイルを選択する
- エディットボックスに直接入力する
- プロジェクトビューのスクリプトファイルタグのコンテキストメニューから"実行"を選択する

ファイル名が入力されている状態でエンターキーを押すと、そのファイルを実行します。（ツールバーの実行ボタンと同等）
 実行されたファイルが正常終了したときに、プロジェクトビューのツリーに登録されていなかった場合は登録されます。
 尚、ファイル名を直接入力する際はフルパスで入力してください。

4.7.3. 標準入出力エディットボックス

Python スクリプトファイル中の `print()` 命令による表示や `input()` 命令による入力、及びスクリプトの直接入力による実行を行います。

スクリプトファイルやコマンドの直接入力には、Python3 の記述と本ソフトウェアが提供しているインターフェース関数を使うことができます。

エディットボックス上では以下の操作が可能です。

Table 37

左右キー	カーソルが左右に移動します。移動範囲はプロンプト（『>>>』もしくは『...』）の右側のみとなります。
上下キー	<ul style="list-style-type: none"> • 上：直接入力したコマンドの履歴を遡ります • 下：遡った履歴を進めます
エンターキー	プロンプトの右側に入力した内容を Python インタプリタに送ります。Python スクリプトの文法に合っていれば実行されます。合っていない場合はエラー情報が出力されます。
Ctrl+C	選択範囲の文字列をクリップボードにコピーします。
Ctrl+V	クリップボードにコピーした文字列をペーストします。Python スクリプトの文法に合わない文字列を貼り付ける場合は注意してください。
マウス左ボタン	エディットボックス上の任意の箇所にカーソルを合わせることや、範囲を選択するために使用します。

4.8. ステータスバー

ウィンドウの一番下に表示され、各選択項目の説明、動作状態、トリガ・カーソル・マーカ間の時間を表示します。

表示位置をマーク 1 に移動	C: 266.60 s	C - M1: 2.10 s	C - M2: -2.48 s	M1: 264.50 s	M1 - M2: -4.58 s	M2: 269.08 s	100.00 ms	...
①	②	③	④	⑤	⑥	⑦	⑧	

Figure 79

- ① 各コマンドの簡易説明を表示します。
- ② カーソルのサンプリング結果の先頭からの時間を表示します。
- ③ カーソル - マーカ 1 間の時間を表示します。
- ④ カーソル - マーカ 2 間の時間を表示します。
- ⑤ マーカ 1 のサンプリング結果の先頭からの時間を表示します。
- ⑥ マーカ 1 - マーカ 2 間の時間を表示します。
- ⑦ マーカ 2 のサンプリング結果の先頭からの時間を表示します。
- ⑧ 1 目盛あたりの時間を表示します。

5. スクリプト

5.1. インターフェース関数一覧

Python スクリプトから本ソフトウェアの各機能を使用するためにインターフェース関数を提供しています。この頁ではインターフェース関数の意味と使用例を機能別で説明しています。

5.1.1. デバイス設定関連

qj_setdevicereference("device", "newname") 機能：デバイス参照名の設定 引数の型：文字列, 文字列 戻り値：なし

指定したデバイスに参照名をつけます。プロジェクトビューの『デバイス参照名の設定』機能と同等です。以降はそのデバイスの元々の名前 (dev n) かこの関数で割り当てた名前でアクセスが可能になります。また、プロジェクトビューのデバイス参照名やデバイスプロパティの参照名もここで設定した名前になります。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたデバイスがボード上に存在しない。

使用例：

```
>>> qj_setdevicereference("dev0", "U1")
>>> a = qj_getpinvalue("U1.pin0")
>>> print (a)
0
```

qj_setpinname("device", "pin", "newname") 機能：ピンの名前を変更 引数の型：文字列, 文字列, 文字列 戻り値：なし

指定デバイスの指定ピンの名前に newname を割り当てます。プロジェクトビューの『ポート名の変更』機能と同等です。以降はそのピンの元々の名前かこの関数で割り当てた名前でアクセスが可能になります。また、プロジェクトビューのポート名もここで設定した名前になります。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例：

```
>>> qj_setpinname("dev0", "pin0", "addr0")
>>> a = qj_getpinvalue("dev0.addr0")
>>> print (a)
0
```

qj_setpinnamefile("device", "filename") 機能：ピンの名前を変更 引数の型：文字列, 文字列 戻り値：なし

指定デバイスのピンの名前を指定したファイルの内容で一斉に変更します。プロジェクトビューの『ポート名の変更』機能のファイル指定と同等です。以降はそのピンの元々の名前かこの関数で割り当てた名前でアクセスが可能になります。また、プロジェクトビューのポート名もここで設定した名前になります。ファイル名はフルパスで指定してください。ただし Python のエスケープシーケンスの仕様上、各ディレクトリの"¥"は二つ重ねてください。

使用例：

```
>>> qj_setpinnamefile("dev0", "C:¥¥Query-J¥¥samplenamefile.txt")
>>> a = qj_getpinvalue("dev0.addr0")
>>> print (a)
0
```

qj_newgroup("group", ["pin", ...], local) 機能：グループ作成 引数の型：文字列, 文字列リスト, int 戻り値：なし

指定された pin リストで group 名のグループを構成します。既に同名のグループがある場合は上書きします。リストの先頭がグループの MSB となります。3 番目の引数が 0 なら作成したグループをプロジェクトビューに登録します。それ以外の場合はプロジェクトビューに登録しません。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_newgroup("GROUP0", list, 0)
>>> qj_setgroupvalue("GROUP0", 5, 7)
>>> vallist = qj_getpinvaluelist(list)
>>> print (vallist)
[1, 0, 1]
```

qj_getgroupmember("group")

機能 : グループの構成ピン名を取得 引数の型 : 文字列 戻り値 : 文字列リスト

指定されたグループを構成しているピン名のリストを返します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_newgroup("GROUP0", list, 0)
>>> namelist = qj_getgroupmember("GROUP0")
>>> print (namelist)
['dev0.pin0', 'dev0.pin1', 'dev0.pin2']
```

qj_setbypass("device", onoff)

機能 : デバイスの BYPASS 設定 引数の型 : 文字列, int 戻り値 : なし

指定したデバイスの BYPASS 設定を行います。onoff の値が 1 ならそのデバイスは次回以降の qj_dosample() や qj_doextest()などのバウンダリスキャン関数で BYPASS として扱われ、デバイスの各ピンの値は更新されません。onoff の値が 0 なら通常のデバイスとして扱われ、デバイスの各ピンの値が更新されます。それ以外の値は、Python の規則にのっとり true(1) として扱います。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたデバイスがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev1.pin0"]
>>> qj_setextestlist(list)
>>> qj_setbypass("dev0", 1)
>>> qj_doextest()
>>> vallist = qj_getpinvaluelist(list)
```

qj_getdeviceproperty("device")

機能 : デバイスのプロパティを取得 引数の型 : なし 戻り値 : dictionary

指定されたデバイスの各種情報を返します。dictionary 型の各キー値は以下の通りです。

Table 38

"ref"	デバイス参照名のキーです。値はデバイスプロパティダイアログの『デバイス参照名』文字列です。
"device"	デバイス名のキーです。値はデバイスプロパティダイアログの『デバイス名』文字列です。
"package"	デバイスパッケージのキーです。値はデバイスプロパティダイアログの『パッケージ』文字列です。
"chain"	バウンダリスキャンチェーン上のデバイス番号のキーです。値はデバイスプロパティダイアログの『JTAG チェーン』の int 値です。
"bsdllib"	BSDL ファイルに記述されているデバイス ID のキーです。値はデバイスプロパティダイアログの『IDCODE(BSDL)』文字列です。
"readid"	IDCODE コマンドでデバイスから読み出した ID のキーです。値はデバイスプロパティダイアログの『IDCODE(デバイス)』文字列です。
"brlen"	デバイスのバウンダリレジスタ長のキーです。値はデバイスプロパティダイアログの『BR 長』の int 値です。
"bsdlir"	BSDL ファイルに記述されているデバイスの IR レジスタ長のキーです。値はデバイスプロパティダイアログの『IR 長(BSDL)』の int 値です。
"readir"	バウンダリスキャンチェーンから読み出したデバイスの IR レジスタ長のキーです。値はデバイスプロパティダイアログの『IR 長(検出/ユーザー設定)』の int 値です。

"bypass"	デバイスの BYPASS 設定のキーです。値はデバイスプロパティダイアログの『BYPASS 設定』を文字列に置き換えたものです。BYPASS 設定時に"true"を返します
----------	--

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたデバイスがボード上に存在しない。

使用例 :

```
>>> dic = qj_getdeviceproperty("dev0")
>>> print (dic)
{'readid': '001A200F', 'chain': 0, 'package': 'TBP_208A', 'bsdlir': 32, 'bsdlib': '001A200F', 'readir': 32,
'brlen': 385, 'bypass': 'false', 'device': 'SH7705', 'ref': 'U1'}
```

5.1.2. 値設定関連

qj_setpinvalue("pin", value)

機能 :ピンの値を設定 **引数の型 :**文字列, int **返り値 :**なし

指定されたピンの値を指定された値にします。主な用途は qj_doeptest()でドライブする際の出力値の設定です。入力専用ピンが指定された場合は何もしません。HIGH_Z(2)を設定した場合は出力禁止になります。特に入出力ピンに HIGH_Z を設定した場合は、以降の qj_getpinvalue()などで 入力値が返ることに注意してください。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1"]
>>> qj_setextestlist(list)
>>> qj_setpinvalue("dev0.pin0", LOW)
>>> qj_doeptest()
>>> vallist = qj_getpinvaluelist(list)
>>> print (vallist)
[0, 0]
```

qj_getpinvalue("pin")

機能 :ピンの値を取得 **引数の型 :**文字列 **返り値 :**int

指定されたピンの値を返します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例 :

```
>>> qj_dosample()
>>> a = qj_getpinvalue("dev0.pin0")
>>> print (a)
0
```

qj_setpinvaluelist(["pin", ...], [value, ...])

機能 :ピンの値を設定 **引数の型 :**文字列リスト, int リスト **返り値 :**なし

指定された 1 つ以上のピンの値をそれぞれ指定された値にします。主な用途は qj_doeptest()でドライブする際の出力値の設定です。入力専用ピンが指定された場合は何もしません。HIGH_Z(2)を設定した場合は出力禁止になります。特に入出力ピンに HIGH_Z を設定した場合は、以降の qj_getpinvalue()などで 入力値が返ることに注意してください。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。
- pin リストと value リストの要素数が異なる。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1"]
>>> vallist = [LOW, HIGH]
>>> qj_setpinvaluelist(list, vallist)
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_setextestlist(list)
>>> qj_doextest()
>>> vallist = qj_getpinvaluelist(list)
>>> print (vallist)
[0, 1, 0]
```

qj_getpinvaluelist(["pin", ...])
機能 : ピンの値を取得 引数の型 : 文字列 返り値 : int リスト

指定されたピンの値を返します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例 :

```
>>> qj_dosample()
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> vlist = qj_getpinvaluelist(list)
>>> print (vlist)
[0, 1, 0]
```

qj_setgroupvalue("group", value, mask)
機能 : グループの値を設定 引数の型 : 文字列, int, int 返り値 : なし

指定されたグループの値を指定された値にします。value 値と mask 値を組み合わせて各ビット位置に該当するピンの LOW, HIGH, HIGH_Z を表現します。主な用途は qj_doextest() でドライブする際の出力値の設定です。グループの構成ピンのうち、入力専用ピンに指定された値は無視されます。mask 値に 0 が設定されているビットは出力禁止 (HIGH_Z) を意味します。特に入出力ピンに HIGH_Z を設定した場合は、以降の qj_getpinvalue() などで 入力値が返ることに注意してください。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_newgroup("GROUP0", list, 0)
>>> qj_setgroupvalue("GROUP0", 5, 7)
>>> vallist = qj_getpinvaluelist(list)
>>> print (vallist)
[1, 0, 1]
```

qj_getgroupvalue("group")
機能 : グループの値を取得 引数の型 : 文字列 返り値 : int, int

指定されたグループの値とマスク値を返します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_newgroup("GROUP0", list, 0)
>>> qj_dosample()
>>> val, mask = qj_getgroupvalue("GROUP0")
>>> print (val, mask)
5 7
```

qj_setvalue(["pin" or "group", ...], [value, ...], [mask, ...])

機能 : ピンやグループの値を設定 **引数の型 :** 文字列リスト, int リスト, int リスト **返り値 :** なし

指定されたピンやグループの値を指定された値にします。指定されたのがピンなら mask 値は無視されて value 値を設定します。指定されたのがグループなら value 値と mask 値を組み合わせて各ビット位置に該当するピンの LOW, HIGH, HIGH_Z を設定します。主な用途は qj_doeptest() でドライブする際の出力値の設定です。入力専用ピンに指定された値は無視されます。HIGH_Z(2)は出力禁止を意味します。特に入出力ピンに HIGH_Z を設定した場合は、以降の qj_getpinvalue() などで入力値が返ることに注意してください。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループやピンがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2"]
>>> qj_newgroup("GROUP0", list, 0)
>>> list = ["GROUP0", "dev0.pin3", "dev0.pin4"]
>>> vallist = [5, LOW, HIGH]
>>> masklist = [7, 1, 1]
>>> qj_setvalue(list, vallist, masklist)
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2", "dev0.pin3", "dev0.pin4"]
>>> vallist = qj_getpinvalue(list)
>>> print (vallist)
[1, 0, 1, 0, 1]
```

qj_getvalue(["pin" or "group", ...])

機能 : ピンやグループの値を取得 **引数の型 :** 文字列リスト **返り値 :** (int, int)リスト

指定されたピンやグループの値とマスク値をリスト形式で取得します。リストの構造は [(値 0, マスク 0), (値 1, マスク 1), ...] のようになります。指定されたのがピンなら mask 値は常に 1 です。指定されたのがグループなら value 値と mask 値を組み合わせて値を表現します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループやピンがボード上に存在しない。

使用例 :

```
>>> qj_dosample()
>>> list = ["GROUP0", "dev0.pin3", "dev0.pin4"]
>>> vlist = qj_getvalue(list)
>>> print (vlist)
[(5, 7), (1, 1), (0, 1)]
```

5.1.3. 接続テスト設定関連

qj_setconnectlist(["output", ...], ["input", ...])

機能 : 接続テスト用のピンを設定 **引数の型 :** 文字列リスト, 文字列リスト **返り値 :** なし

次回以降の qj_dodeconnecttest() で使用するピン名やグループ名のリストに引数で指定したピン名やグループ名を設定します。第一引数には出力側のピンやグループの名前をリストで渡してください。第二引数には入力側のピンやグループの名前をリストで渡してください。この関数を呼ぶたびにそれまでの設定情報は破棄されます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。
- 出力側と入力側に同じピンがある。

使用例 :

```
>>> outlist = ["dev0.out0"]
>>> inlist = ["dev0.in0"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_addconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
```

qj_addconnectlist(["output", ...], ["input", ...])

機能 :接続テスト用のピンを追加 **引数の型 :**文字列リスト, 文字列リスト **返り値 :**なし

次回以降の qj_doconnecttest()で使用するピン名やグループ名のリストに引数で指定したピン名やグループ名を追加します。第一引数には出力側のピンやグループの名前をリストで渡してください。第二引数には入力側のピンやグループの名前をリストで渡してください。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。
- 設定済みのリストと併せて、出力側と入力側に同じピンがある。

使用例 :

```
>>> outlist = ["dev0.out0"]
>>> inlist = ["dev0.in0"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_addconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
```

qj_getconnectlist()

機能 :接続テスト用のピンを取得 **引数の型 :**なし **返り値 :**文字列リスト, 文字列リスト

次回以降の qj_doconnecttest()で使用するピン名やグループ名のリストを返します。一番目の返り値には出力側のリストが返ります。二番目の返り値には入力側のリストが返ります。

使用例 :

```
>>> outlist = ["dev0.out0"]
>>> inlist = ["dev0.in0"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_addconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
```

qj_removeconnectlist(["output", ...], ["input", ...])

機能 :接続テスト用のピンを除外 **引数の型 :**文字列リスト, 文字列リスト
返り値 :なし

次回以降の qj_doconnecttest()で使用するピン名やグループ名のリストから引数で指定したピン名やグループ名を除外します。第一引数には出力側のピンやグループの名前をリストで渡してください。第二引数には入力側のピンやグループの名前をリストで渡してください。

使用例 :

```
>>> outlist = ["dev0.out0", "dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in0", "dev0.in1", "dev0.in2"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out0"]
>>> inlist = []
>>> qj_removeconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
```

qj_clearconnectlist()

機能 :接続テスト用のピンをクリア **引数の型 :**なし **返り値 :**なし

次回以降の qj_doconnecttest()で使用するピン名やグループ名のリストを空にします。

使用例 :

```
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_setconnectlist(outlist, inlist)
>>> vallist = [LOW, LOW, 0x0]
>>> masklist = [1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW]
>>> masklist = [1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
>>> qj_clearconnectlist()
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
[] []
```

qj_setexpectedpinvalue(["pin", ...], [value, ...]) 機能 : ピンの期待値を設定 引数の型 : 文字列リスト, intリスト 戻り値 : なし

指定されたピンに qj_doconnecttest() で結果を判定するために使う期待値を設定します。この関数で指定するピン名は qj_setconnectlist() の入力側に指定されている必要があります。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。
- 指定されたピンが接続テストの入力側リストに存在しない。

使用例 :

```
>>> outlist = ["dev0.out0"]
>>> inlist = ["dev0.in0"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_addconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> qj_setexpectedpinvalue(inlist, vallist)
>>> result, errlist = qj_doconnecttest()
```

qj_setexpectedvalue(["pin" or "group", ...], [value, ...], [mask, ...]) 機能 : ピンやグループの期待値を設定 引数の型 : 文字列リスト, intリスト, intリスト 戻り値 : なし

指定されたピンやグループに qj_doconnecttest() で結果を判定するために使う期待値を設定します。指定されたのがピンなら mask 値は無視され value 値を設定します。指定されたのがグループなら value 値と mask 値を組み合わせて各ビット位置に該当するピンの LOW, HIGH, HIGH_Z を設定します。この関数で指定するピン名やグループ名は qj_setconnectlist() の入力側に指定されている必要があります。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたグループやピンがボード上に存在しない。
- 指定されたグループやピンが接続テストの入力側リストに存在しない。

使用例 :

```
>>> outlist = ["dev0.out0"]
>>> inlist = ["dev0.in0"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_addconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
```

5.1.4. EXTEST 設定関連

qj_setextestlist(["pin or group", ...])
機能 : EXTEST 用のピンを設定 引数の型 : 文字列リスト 戻り値 : なし

次回以降の qj_doeptest() で使用するピン名やグループ名のリストに引数で指定したピン名やグループ名を設定します。この関数を呼ぶたびにそれまでの設定情報は破棄されます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin1", "dev0.pin2", "GROUP0"]
>>> qj_setextestlist(list)
>>> list = qj_getextestlist()
>>> print (list)
['dev0.pin1', 'dev0.pin2', 'GROUP0']
>>> qj_doeptest()
```

qj_addextestlist(["pin or group", ...])
機能 : EXTEST 用のピンを追加 引数の型 : 文字列リスト 戻り値 : なし

次回以降の qj_doeptest() で使用するピン名やグループ名のリストに引数で指定したピン名やグループ名を追加します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。

使用例 :

```
>>> list = ["dev0.pin0"]
>>> qj_setextestlist(list)
>>> list = ["dev0.pin1", "dev0.pin2", "GROUP0"]
>>> qj_addextestlist(list)
>>> list = qj_getextestlist()
>>> print (list)
['dev0.pin0', 'dev0.pin1', 'dev0.pin2', 'GROUP0']
>>> qj_doeptest()
```

qj_getextestlist()
機能 : EXTEST 用のピンを取得 引数の型 : なし 戻り値 : 文字列リスト

次回以降の qj_doeptest() で使用するピン名やグループ名のリストを返します。

使用例 :

```
>>> list = ["dev0.pin1", "dev0.pin2", "GROUP0"]
>>> qj_setextestlist(list)
>>> list = qj_getextestlist()
>>> print (list)
['dev0.pin1', 'dev0.pin2', 'GROUP0']
>>> qj_doextest()
```

qj_removeextestlist(["pin or group", ...])
機能 : EXTEST 用のピンを除外 引数の型 : 文字列リスト 戻り値 : なし

次回以降の qj_doextest() で使用するピン名やグループ名のリストから引数で指定したピン名やグループ名を除外します。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "dev0.pin2", "GROUP0"]
>>> qj_setextestlist(list)
>>> list = ["dev0.pin0"]
>>> qj_removeextestlist(list)
>>> list = qj_getextestlist()
>>> print (list)
['dev0.pin1', 'dev0.pin2', 'GROUP0']
>>> qj_doextest()
```

qj_clearextestlist()
機能 : EXTEST 用のピンをクリア 引数の型 : なし 戻り値 : なし

次回以降の qj_doextest() で使用するピン名やグループ名のリストを空にします。

使用例 :

```
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_setextestlist(list)
>>> qj_doextest()
>>> qj_clearextestlist()
>>> list = qj_getextestlist()
>>> print (list)
[]
```

5.1.5. バウンダリスキャンテスト実行関連

qj_disconnecttest()
機能 : 接続テスト実行 引数の型 : なし 戻り値 : int, 文字列リスト

あらかじめ qj_setconnectlist() で指定したピン及びグループを使って バウンダリスキャンテストの EXTEST コマンドを結果の比較付きで実行します。 実行した結果、入力側に指定した各ピンの値が qj_setexpectedvalue() で設定した期待値と等しければ 一番目の返り値に 1 (true) を返し、二番目の返り値に空の文字列が一つだけ入れられたリストを返します。 実行した結果、どれか一つでも期待値と異なるピンがあれば 一番目の返り値に 0 (false) を返し、二番目の返り値に期待値と異なったピン名のリストを返します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- qj_setconnectlist() や qj_addconnectlist() で (入力側) ピンを登録していない
- 入力側のピンやグループの中に期待値が設定されていないものがある。
- 波形表示ウィンドウが計測中

使用例 :

```
>>> outlist = ["dev0.out0", "dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in0", "dev0.in1", "dev0.in2"]
>>> qj_setconnectlist(outlist, inlist)
>>> outlist, inlist = qj_getconnectlist()
>>> print (outlist, inlist)
['dev0.out0', 'dev0.out1', 'dev0.out2', 'OUTGROUP0'] ['dev0.in0', 'dev0.in1', 'dev0.in2']
>>> vallist = [LOW, LOW, LOW, 0x0]
>>> masklist = [1, 1, 1, 0xF]
>>> qj_setvalue(outlist, vallist, masklist)
>>> vallist = [LOW, LOW, LOW]
>>> masklist = [1, 1, 1]
>>> qj_setexpectedvalue(inlist, vallist, masklist)
>>> result, errlist = qj_doconnecttest()
>>> print (result, errlist)
0 ['dev0.in0']
```

qj_doeptest()

機能 : バウンダリスキャンの EXTEST コマンド実行 **引数の型 :** なし **返り値 :** なし

あらかじめ qj_setextestlist() や qj_addextestlist() で指定したピンを バウンダリスキャンの EXTEST コマンドで操作します。 HIGH や LOW を設定した出力ピンや入出力ピンであればドライブし、入力ピンや HI_Z を設定した入出力ピンであれば状態を取り込みます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- qj_setextestlist() や qj_addextestlist() でピンを登録していない
- 波形表示ウィンドウが計測中

使用例 :

```
>>> list = ["dev0.pin1", "dev0.pin2", "GROUP0"]
>>> qj_setextestlist(list)
>>> list = qj_getextestlist()
>>> print (list)
['dev0.pin1', 'dev0.pin2', 'GROUP0']
>>> qj_doeptest()
```

qj_dosample()

機能 : バウンダリスキャンの SAMPLE コマンド実行 **引数の型 :** なし **返り値 :** なし

ボード上の全てのピンの値をバウンダリスキャンの SAMPLE コマンドで更新します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。

使用例 :

```
>>> qj_dosample()
>>> a = qj_getpinvalue("dev0.pin0")
>>> print (a)
0
```

5.1.6. 波形表示ウィンドウ操作関連

qj_wv_open("name")

機能 : 波形表示パターンを開く **引数の型 :** 文字列 **返り値 :** なし

引数で指定した名前の波形表示パターンを開きます。指定した名前がプロジェクトビューになければ新規作成します。波形表示ウィンドウが閉じていれば開きます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。

使用例 :

```
>>> qj_wv_open("wave 1")
```

qj_wv_close()
機能：波形表示ウィンドウを閉じる 引数の型：なし 戻り値：なし

波形表示ウィンドウを閉じます。既に閉じている場合は何もしません。

使用例：

```
>>> qj_wv_open("wave 1")
>>> qj_wv_close()
```

qj_wv_add(["pin" or "group", ...])
機能：波形表示ウィンドウに信号を追加 引数の型：文字列リスト 戻り値：なし

指定されたピンやグループを現在開いている波形表示ウィンドウ（パターン）に追加します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。
- 波形表示ウィンドウが閉じている。
- 波形表示ウィンドウが計測中

使用例：

```
>>> qj_wv_open("wave 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_wv_add(list)
```

qj_wv_remove(["pin" or "group", ...])
機能：波形表示ウィンドウから信号を除外 引数の型：文字列リスト 戻り値：なし

指定されたピンやグループを現在開いている波形表示ウィンドウ（パターン）から除外します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 波形表示ウィンドウが閉じている。
- 波形表示ウィンドウが計測中

使用例：

```
>>> qj_wv_open("wave 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_wv_add(list)
>>> list = ["dev0.pin0"]
>>> qj_wv_remove(list)
```

qj_wv_setinterval(interval)
機能：サンプリング間隔を設定 引数の型：int 戻り値：int

現在開いている波形表示ウィンドウ（パターン）のサンプリング間隔を設定します。値の単位はミリ秒（ms）です。波形表示ウィンドウのサンプリング間隔のコンボボックスにない値が指定された場合は、指定値以上で最も近い値を設定します。実際に設定された返り値として返します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 波形表示ウィンドウが閉じている。
- 波形表示ウィンドウが計測中。

使用例：

```
>>> qj_wv_open("wave 1")
>>> a = qj_wv_getinterval()
>>> print (a)
100
>>> a = qj_wv_setinterval(250)
>>> print (a)
300
```

qj_wv_getinterval()
機能：サンプリング間隔を取得 引数の型：なし 戻り値：int

現在開いている波形表示ウィンドウ（パターン）のサンプリング間隔を取得します。値の単位はミリ秒（ms）です。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 波形表示ウィンドウが閉じている。

使用例：

```
>>> qj_wv_open("wave 1")
>>> a = qj_wv_getinterval()
>>> print (a)
100
```

qj_wv_start()
機能 : サンプリング開始 引数の型 : なし 戻り値 : なし

現在開いている波形表示ウィンドウ（パターン）の計測を開始します。既に計測中であれば何もしません。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 波形表示ウィンドウが閉じている。
- 信号が一つも登録されていない。

使用例 :

```
>>> qj_wv_open("wave 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_wv_add(list)
>>> qj_wv_start()
```

qj_wv_stop()
機能 : サンプリング停止 引数の型 : なし 戻り値 : なし

現在開いている波形表示ウィンドウ（パターン）の計測を停止します。既に停止状態であれば何もしません。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 波形表示ウィンドウが閉じている。

使用例 :

```
>>> qj_wv_open("wave 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_wv_add(list)
>>> qj_wv_start()
>>> input("Wait Input: ")
Wait Input:
>>> qj_wv_stop()
```

5.1.7. 端子状態一覧ウィンドウ操作関連

qj_pv_open("name")
機能 : 端子状態一覧リストを開く 引数の型 : 文字列 戻り値 : なし

引数で指定した名前の端子状態一覧リストを開きます。指定した名前がプロジェクトビューになければ新規作成します。端子状態一覧ウィンドウが閉じていれば開きます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。

使用例 :

```
>>> qj_pv_open("list 1")
```

qj_pv_close()
機能 : 端子状態一覧ウィンドウを閉じる 引数の型 : なし 戻り値 : なし

端子状態一覧ウィンドウを閉じます。既に閉じている場合は何もしません。

使用例 :

```
>>> qj_pv_open("list 1")
>>> qj_pv_close()
```

qj_pv_add(["pin" or "group", ...])
機能 : 端子状態一覧ウィンドウに信号を追加 引数の型 : 文字列リスト 戻り値 : なし

指定されたピンやグループを現在開いている端子状態一覧ウィンドウ（リスト）に追加します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンやグループがボード上に存在しない。
- 端子状態一覧ウィンドウが閉じている。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
```

qj_pv_remove(["pin" or "group", ...])
機能 : 端子状態一覧ウィンドウからピンを除外 引数の型 : 文字列リスト 戻り値 : なし

指定されたピンやグループを現在開いている端子状態一覧ウィンドウ（リスト）から除外します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> list = ["dev0.pin0"]
>>> qj_pv_remove(list)
```

qj_pv_setpinvalue("pin", value)

機能 : ピンの値を設定 **引数の型 :** 文字列, int **返り値 :** なし

指定されたピンの値を指定された値にします。主な用途は qj_pv_doeextest()でドライブする際の出力値の設定です。入力専用ピンが指定された場合は何もしません。HIGH_Z(2)を設定した場合は出力禁止になります。実行後に端子状態一覧ウィンドウの表示を更新します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。
- 指定されたピンがボード上に存在しない。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> qj_pv_setpinvalue("dev0.pin0", LOW)
```

qj_pv_setpinvaluelist(["pin", ...], [value, ...])

機能 : ピンの値を設定 **引数の型 :** 文字列リスト, int リスト **返り値 :** なし

指定された 1 つ以上のピンの値をそれぞれ指定された値にします。主な用途は qj_pv_doeextest()でドライブする際の出力値の設定です。入力専用ピンが指定された場合は何もしません。HIGH_Z(2)を設定した場合は出力禁止になります。実行後に端子状態一覧ウィンドウの表示を更新します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。
- 指定されたピンがボード上に存在しない。
- pin リストと value リストの要素数が異なる。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> list = ["dev0.pin0", "dev0.pin1"]
>>> vallist = [LOW, HIGH]
>>> qj_pv_setpinvaluelist(list, vallist)
```

qj_pv_dosample()

機能 : バンダリスキヤンの SAMPLE コマンド実行 **引数の型 :** なし **返り値 :** なし

端子状態一覧ウィンドウ上で SAMPLE コマンドを実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。
- 端子状態一覧リストにピンが登録されていない。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> qj_pv_dosample()
```

qj_pv_doeextest()

機能 : バンダリスキヤンの EXTEST コマンド実行 **引数の型 :** なし **返り値 :** なし

端子状態一覧ウィンドウ上で EXTEST コマンドを実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。
- 端子状態一覧リストにピンが登録されていない。

- 波形表示ウィンドウが計測中

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> qj_pv_setpinvalue("dev0.pin0", LOW)
>>> qj_pv_doeextest()
```

qj_pv_savecsv()

機能 : 端子状態一覧リストの保存 引数の型 : なし 戻り値 : なし

端子状態一覧リストの状態を CSV ファイルに保存します。端子状態一覧ウィンドウの『CSV 保存』ボタン押下と同等です。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 端子状態一覧ウィンドウが閉じている。

使用例 :

```
>>> qj_pv_open("list 1")
>>> list = ["dev0.pin0", "dev0.pin1", "GROUP0"]
>>> qj_pv_add(list)
>>> qj_pv_dosample()
>>> qj_pv_savecsv()
```

5.1.8. 接続テストウィンドウ関連

qj_cv_open("name")

機能 : 接続テストウィンドウを開く 引数の型 : 文字列 戻り値 : なし

引数で指定した名前の接続テストを開きます。指定した名前がプロジェクトビューになければ新規作成します。接続テストウィンドウが閉じていれば開きます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。

使用例 :

```
>>> qj_cv_open("test 1")
```

qj_cv_close()

機能 : 接続テストウィンドウを閉じる 引数の型 : なし 戻り値 : なし

接続テストウィンドウを閉じます。既に閉じている場合は何もしません。

使用例 :

```
>>> qj_cv_open("test 1")
>>> qj_cv_close()
```

qj_cv_addconnectlist(["output", ...], ["input", ...])

**機能 : 接続テストウィンドウにネットを追加 引数の型 : 文字列リスト,
文字列リスト 戻り値 : なし**

接続テストウィンドウにネットを追加し、引数で指定した出力ピンリストと入力ピンリストを設定します。第一引数には出力側のピンやグループの名前をリストで渡してください。第二引数には入力側のピンやグループの名前をリストで渡してください。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 指定したピンやグループがボード上に存在しない
- 出力ピンリストと入力ピンリストの両方に同じ名前のピンやグループが指定されている。
- 出力ピンリストに入力専用ピンが含まれている。
- 入力ピンリストに出力専用ピンが含まれている。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
```

qj_cv_removenet(netnumber)

機能 :接続テストウィンドウのネットを削除 **引数の型 :**int **返り値 :**なし

指定された番号のネットを接続テストウィンドウから削除します。存在しない番号を指定した場合は何もしません。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_removenet(1)
```

qj_cv_setoutput(netnumber, [value, ...])

機能 :ネットに出力値を設定 **引数の型 :**int, intリスト **返り値 :**なし

接続テストウィンドウ上の指定された番号のネットに出力値を設定します。それまでの出力値は上書きされます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> qj_cv_setoutput(1, values)
```

qj_cv_addoutput(netnumber, [value, ...])

機能 :ネットに出力値を追加 **引数の型 :**int, intリスト **返り値 :**なし

接続テストウィンドウ上の指定された番号のネットに出力値を追加します。それまでの出力値の下の行に追加されます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1]
>>> qj_cv_setoutput(1, values)
>>> values = [2, 3]
>>> qj_cv_addoutput(1, values)
```

qj_cv_editoutput(netnumber, valueindex, value)

機能 :ネットに出力値を変更 **引数の型 :**int, int, int **返り値 :**なし

接続テストウィンドウ上の指定された番号のネットの指定されたインデックスの出力値を変更します。各ネットのインデックスはそれぞれ0から始まります。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 存在しないインデックスが指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> qj_cv_setoutput(1, values)
>>> qj_cv_editoutput(1, 1, 3)
```

qj_cv_removeoutput(netnumber, valueindex)
機能 : ネットから出力値を削除 引数の型 : int, int 戻り値 : なし

接続テストウィンドウ上の指定された番号のネットの指定されたインデックスの出力値を削除します。各ネットのインデックスはそれぞれ 0 から始まります。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 存在しないインデックスが指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> qj_cv_setoutput(1, values)
>>> qj_cv_removeoutput(1, 1)
```

qj_cv_setexpected(netnumber, valueindex, value, mask)
機能 : ネットに期待値を設定 引数の型 : int, int, int, int
戻り値 : なし

接続テストウィンドウ上の指定された番号のネットの指定されたインデックスの出力値に対する期待値を設定します。各ネットのインデックスはそれぞれ 0 から始まります。値とマスク値を組み合わせて最終的な値が決まります。マスクビットが 0 のとき、そのビットは接続テストでの値比較の結果を無視します。（常に OK として扱います。接続テストウィンドウ上では文字"x"が表示されます）

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 存在しないインデックスが指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> qj_cv_setoutput(1, values)
>>> qj_cv_setexpected(1, 0, 0b00, 0b11)
>>> qj_cv_setexpected(1, 1, 0b01, 0b01)
```

qj_cv_setexpectedlist(netnumber, [value, ...], [mask, ...])
機能 : ネットに期待値を設定 引数の型 : int, int リスト,
int リスト 戻り値 : なし

接続テストウィンドウ上の指定された番号のネットの各出力値に対する期待値を設定します。値とマスク値を組み合わせて最終的な値が決まります。マスクビットが 0 のとき、そのビットは接続テストでの値比較の結果を無視します。（常に OK として扱います。接続テストウィンドウ上では文字"x"が表示されます） 出力値の個数より多くの期待値リストが引数に渡された場合、余剰分は無視されます。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 期待値リストとマスク値リストの個数が異なる。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> qj_cv_setoutput(1, values)
>>> values = [0, 1, 2, 3]
>>> masks = [0b11, 0b01, 0b11, 0b10]
>>> qj_cv_setexpectedlist(1, values, masks)
```

qj_cv_setvalue(netnumber, [output, ...], [expect, ...], [mask, ...])

機能 : ネットに出力値と期待値を設定 引数の型 : int, int リスト, int リスト, int リスト 戻り値 : なし

接続テストウィンドウ上の指定された番号のネットに出力値と期待値を設定します。期待値は **expect** と **mask** を組み合わせて最終的な値が決まります。マスクビットが 0 のとき、そのビットは接続テストでの値比較の結果を無視します。（常に OK として扱います。接続テストウィンドウ上では文字"x"が表示されます） それまでの出力値及び期待値の設定は上書きされます。出力値の個数より多くの期待値リストが引数に渡された場合、余剰分は無視されます。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 期待値リストとマスク値リストの個数が異なる。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> values = [0, 1, 2, 3]
>>> expects = [0, 1, 2, 3]
>>> masks = [0b11, 0b01, 0b11, 0b10]
>>> qj_cv_setvalue(1, values, expects, masks)
```

qj_cv_autovalue(netnumber or NULL)

機能 : 出力パターンの自動生成 引数の型 : int 戻り値 : なし

接続テストウィンドウ上の指定された番号のネットに対して「出力パターンの自動生成」メニューを実行します。番号を省略した場合は全てのネットに対して実行します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_autovalue()
```

qj_cv_learnexpected(netnumber, valueindex) or (NULL)
機能：期待値の学習 引数の型：int, int 返り値：なし

接続テストウィンドウ上の指定された番号のネットの指定されたインデックスの出力値に対して「期待値の学習」メニューを実行します。各ネットのインデックスはそれぞれ 0 から始まります。引数を全て省略した場合は全てのネットの全ての出力値に対して実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 存在しないインデックスが指定された。

使用例：

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_autovalue()
>>> qj_cv_learnexpected(1, 0)
```

qj_cv_dotest(netnumber, valueindex) or (NULL)
機能：接続テストの実行 引数の型：int, int 返り値：なし

接続テストウィンドウ上の指定された番号のネットの指定されたインデックスの出力値に対して「接続テスト」メニューを実行します。各ネットのインデックスはそれぞれ 0 から始まります。引数を全て省略した場合は全てのネットの全ての出力値に対して実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。
- 存在しないネット番号が指定された。
- 存在しないインデックスが指定された。
- 指定されたインデックスは出力値か期待値の少なくともどちらかが設定されていない。

使用例：

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_autovalue()
>>> qj_cv_learnexpected(1, 0)
>>> qj_cv_dotest()
```

qj_cv_opennetlist()
機能：ネットリストを開く 引数の型：なし 返り値：なし

接続テストウィンドウの「ネットリストを開く」メニューを実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている

使用例：

```
>>> qj_cv_open("test 1")
>>> qj_cv_opennetlist()
```

qj_cv_savenetlist()
機能：ネットリストの保存 引数の型：なし 返り値：なし

接続テストウィンドウの「ネットリストの保存」メニューを実行します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_savenetlist()
```

qj_cv_saveresult()
機能 : 接続テスト結果の保存 引数の型 : なし 戻り値 : なし

接続テストウィンドウの「接続テスト結果の保存」メニューを実行します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- 整合性テストに通っていない。
- 接続テストウィンドウが閉じている。
- 接続テストウィンドウにネットが一つも登録されていない。

使用例 :

```
>>> qj_cv_open("test 1")
>>> outlist = ["dev0.out1", "dev0.out2", "OUTGROUP0"]
>>> inlist = ["dev0.in1", "dev0.in2"]
>>> qj_cv_addconnectlist(outlist, inlist)
>>> qj_cv_autovalue()
>>> qj_cv_learnexpected(1, 0)
>>> qj_cv_dotest()
>>> qj_cv_saveresult()
```

5.1.9. 環境設定関連

qj_getfrequencylist()
機能 : 周波数リストを取得 引数の型 : なし 戻り値 : 文字列リスト

バウンダリスキャンテストを行う周波数として設定できる値を返します。

以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- プローブと接続されていない

使用例 :

```
>>> list = qj_getfrequencylist()
>>> print (list)
['Auto', '[2...624]kHz', '0.62MHz', '0.93MHz', '1.25MHz', '1.87MHz', '2.50MHz', '3.75MHz', '5.00MHz',
'7.50MHz', '10.00MHz', '15.00MHz']
>>> a = qj_setfrequency("5.00MHz")
>>> print (a)
5000000
```

qj_setfrequency("frequency")
機能 : バウンダリスキャン時の周波数設定 引数の型 : 文字列 戻り値 : int

バウンダリスキャンテストを行う際の周波数を文字列で設定します。返り値として実際に設定された周波数の値を int 値で返します。以下の場合は標準出力にエラー情報を出力し、スクリプトファイル実行中なら中断します。

- プローブと接続されていない

使用例 :

```
>>> list = qj_getfrequencylist()
>>> print (list)
['Auto', '[2...624]kHz', '0.62MHz', '0.93MHz', '1.25MHz', '1.87MHz', '2.50MHz', '3.75MHz', '5.00MHz',
'7.50MHz', '10.00MHz', '15.00MHz']
>>> a = qj_setfrequency("5.00MHz")
>>> print (a)
5000000
```

qj_getfrequency()
機能：パワーダリスキヤン時の周波数を取得 引数の型：なし 戻り値：int

パワーダリスキヤンテストを行う際の周波数を返します。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- プローブと接続されていない

使用例：

```
>>> list = qj_getfrequencylist()
>>> print (list)
['Auto', '[2...624]kHz', '0.62MHz', '0.93MHz', '1.25MHz', '1.87MHz', '2.50MHz', '3.75MHz', '5.00MHz',
'7.50MHz', '10.00MHz', '15.00MHz']
>>> a = qj_getfrequency()
>>> print (a)
5000000
```

5.1.10. その他

qj_runthefile("filename" or NULL)
機能：スクリプトファイル実行 引数の型：文字列 戻り値：なし

引数で指定されたスクリプトファイルを実行します。スクリプトペインの『実行ボタン』機能と同等です。引数を省略した場合はスクリプトペインの『ファイル名』エディットボックスに入力されているファイルを実行します。引数にファイル名が渡されている場合はそのファイルを実行します。尚、ファイル名はフルパスで指定してください。ただし Python のエスケープシーケンスの仕様上、各ディレクトリの "¥" は二つ重ねてください。存在しないファイル名が指定された場合は、スクリプトペインの『実行ボタン』を押したときと同様にファイルが存在しない旨のメッセージボックスが表示されます。

使用例：

```
>>> qj_runthefile("C:¥¥Query-J¥¥samplescript.py")
>>> qj_runthefile()
```

qj_setalias("pin", "newname")
機能：ピン名のエイリアス設定 引数の型：文字列, 文字列 戻り値：なし

pin で指定されたピンに newname を割り当てます。qj_getvalue()などでデバイス名の記述を省略したいピンがあれば使用します。登録済みの newname を設定した場合は上書きされます。

以下の場合は標準出力にエラー情報を出し、スクリプトファイル実行中なら中断します。

- 指定されたピンがボード上に存在しない。

使用例：

```
>>> qj_setalias("dev0.pin0", "P0")
>>> a = qj_getpinvalue("P0")
>>> print (a)
0
```

5.2. インターフェース関数使用時の注意

本ソフトウェアが提供する Python スクリプト用のインターフェース関数を使う場合は以下の点にご注意ください。

5.2.1. 関数の呼び出し方について

全てのインターフェース関数は queryj モジュールで定義されており、本ソフトウェア起動時に Python インタプリタにインポートさせています ("import queryj")。

併せて全てのインターフェース関数をグローバル関数として登録しています ("from queryj import *")。

よって、全てのインターフェース関数は queryj.qj_xxxx() もしくは qj_xxxx() と記述してコールすることができます。

本ソフトウェアの起動中に queryj モジュールを削除しないでください。削除した場合、以後のスクリプト実行機能の動作保証はいたしません。

5.2.2. ピン名の指定について

Python スクリプト上でピン名を指定する際は以下のように記述してください。

"デバイス名.ピン名 (or ポート名)"

(例 : int 型の変数『value』に『dev0 デバイス』の『1 番ピン』の値を取得する場合 : value =
qj_getpinvalue("dev0.1"))

(例 : 文字列リスト型の変数『list』に『dev0 デバイス』の『1 番ピン』の名前をセットする場合 : list = ["dev0.1"]

ただし、qj_setpinname()だけは例外としてデバイス名とピン名を分けて指定します。

(例 : 『dev0 デバイス』の『1 番ピン』に『data0』という名前を付ける場合 : qj_setpinname("dev0", "1", "data0"))
qj_setpinname() や『ポート名の変更』で名前を変更した後は変更後の名前かピン名で指定できますが、変更前の名前では指定できなくなります。つまり、プロジェクトビューなどで現在表示されているピン名かポート名でのみ指定ができます。

5.2.3. ピンの値について (データフォーマット)

・ピン単体での使用

```
0 >> LOW  
1 >> HIGH  
2 >> HI_Z (ハイインピーダンス)
```

本ソフトウェアからは上記の値で LOW, HIGH, HI_Z という変数を併せて提供しています。

(例 : 『dev0 デバイス』の『1 番ピン』に『LOW』を設定する場合 : qj_setpinvalue("dev0.1", LOW))

・グループでの使用

ビットマスクと組み合わせて使用します。

```
0 (マスク 1) >> Low  
1 (マスク 1) >> High  
0 (マスク 0) >> High_Z  
1 (マスク 0) >> High_Z
```

例えば値=0xAA、マスク値=0xF0 なら、各ビットは左から順に
High, Low, High, Low, High_Z, High_Z, High_Z, High_Z となります。

(例 : list = ["GROUP0"]

```
vlist = [0xaa]  
mlist = [0xf0]  
qj_setvalue(list, vlist, mlist) )
```

5.2.4. ピンの値について（タイプ別の設定/取得）

`qj_setpinvalue()`・`qj_getpinvalue()`・`qj_setgroupvalue()`・`qj_getgroupvalue()`・`qj_setvalue()`・`qj_getvalue()`といった、値の設定や取得を行う関数での値の取り扱いは対象のピンの種類に影響を受けてますのでご注意ください。

Table 39

入力専用ピン	・set 関数では何もしません。get 関数では入力側の値を返します。
出力専用ピン	・set 関数では出力値を設定します。get 関数でも出力側の値を返します。
入出力ピン	・set 関数では出力値を設定します。 ・set 関数で HIGH か LOW を設定した場合は get 関数で出力側の値を返します。 ・set 関数で HI_Z を設定した場合は get 関数で入力側の値を返します。

上記のことから、あるピンに対して set 関数と get 関数を行った際に set 値と get 値が一致しないことがあります。

例：入力専用ピンに対して LOW を設定したが HIGH を取得した。

例：入出力ピンに対して HI_Z を設定したら LOW を取得した。

etc.

以上のことからグループ関数（`qj_setgroupvalue()`・`qj_getgroupvalue()`）での値とマスク値の設定と取得の関係を改めて表にまとめると以下のようになります。

Table 40

ピン種類	set(値、マスク)	get(値、マスク)
入力専用ピン	LOW(0, 1) ※但し何もしない	LOW(0, 1) or HIGH(1, 1) ※最後に SAMPLE した状態
	HIGH(1, 1) ※但し何もしない	LOW(0, 1) or HIGH(1, 1) ※最後に SAMPLE した状態
	HI_Z(0, 0) ※但し何もしない	LOW(0, 1) or HIGH(1, 1) ※最後に SAMPLE した状態
出力専用ピン	LOW(0, 1)	LOW(0, 1)
	HIGH(1, 1)	HIGH(1, 1)
	HI_Z(0, 0)	HI_Z(0, 0)
入出力ピン	LOW(0, 1)	LOW(0, 1)
	HIGH(1, 1)	HIGH(1, 1)
	HI_Z(0, 0)	LOW(0, 1) or HIGH(1, 1) ※最後に SAMPLE した状態

5.2.5. ピンの値について（ドライブ・更新のタイミング）

ターゲットボード上の信号をドライブしたり、本ソフトウェアに取り込まれるタイミングは `qj_doeptest()`・`qj_disconnecttest()`・`qj_dosample()`といった バウンダリスキャンテスト実行関連の関数が実行されたときです。

例えば、`qj_setpinvalue()` などの値設定関連の関数で値を設定した時点ではターゲットボード上の信号はドライブされません。
(本ソフトウェア上では HIGH となっていてもターゲットボード上では LOW といった状態が起こります。)

Table 41

ピン種類	set(値、マスク)
qj_doeptest()	<p>実行時に、qj_getextestlist()で取得されるピンのみ操作します。 ピンの種類によって以下のように動作します。</p> <p>入力専用ピン：状態を取り込みます。</p> <p>出力専用ピン：qj_setpinvalue()などで設定された値をドライブします。</p> <p>入出力ピン：qj_setpinvalue()などで HIGH か LOW が設定されていればドライブします（出力動作）。 HI_Z が設定されていれば状態を取り込みます（入力動作）。</p>
qj_disconnecttest()	実行時に、qj_getconnectlist()で取得されるピンのみ操作します。出力側に登録されているピンを qj_setpinvalue()などで設定された値にドライブし、入力側に登録されているピンの状態を取り込みます。
qj_dosample()	全てのピンの状態を取り込みます。出力信号のドライブは行いません。

qj_doeptest() や qj_disconnecttest() 時に出力専用ピン及び入出力ピンをドライブするために、そのピンに対して事前に qj_setpinvalue などの値設定用の関数を適用する必要があります。
(出力値を設定することで端子状態ウィンドウで言うところの"チェックを付ける"状態になります。)

5.2.6. デバイス名について

デバイス名の初期値は"devN"（N は 0 から始まるバウンダリスキヤンチェーン上のデバイス順）となります。
そのため qj_setdevicereference() では、大文字小文字の区別なく新しい参照名として"devN"文字列を設定することを禁止しています。

5.2.7. クロック設定について

qj_setfrequency() で指定する周波数について、例えば以下のような文字列の使用を許可しています。

Table 42

"Auto"	例：qj_setfrequency("auto") >> 8.33MHz
単位なしの数値	例：qj_setfrequency("10000") >> 低速クロック 10kHz
k(Hz)付き数値	例：qj_setfrequency("100k") >> 低速クロック 100kHz qj_setfrequency("100kHz") >> 低速クロック 100kHz
M(Hz)付き数値	例：qj_setfrequency("12M") >> 10MHz qj_setfrequency("10MHz") >> 10MHz

キロ、メガ以外の単位はプローブがその単位のクロックを出力できないため許可していません。

A. 付録

A.1. 各種デバイス接続 対応方法

本ソフトウェア、プローブ、及びターゲットボードとの特殊な接続の事例を説明します。

A.1.1. Vref 信号が接続されていないターゲット

プローブの JTAG コネクタにある Vref 信号に繋ぐ信号がターゲットボード側にない場合に、プローブから JTAG 信号の High レベルとして 3.3V を出力するようにします（例:SH7705 など）。

* SH7705 は（株）ルネサステクノロジの商標です。



Vref 信号が接続されている場合は、この設定は必要ありません。

1) プロジェクトの新規作成

ツールボタン	メニューバーの操作	ショートカットキー
	ファイル → プロジェクトの新規作成	Ctrl + N

上記メニューを実行すると、プロジェクトビューに「無題」プロジェクトが作成されます。

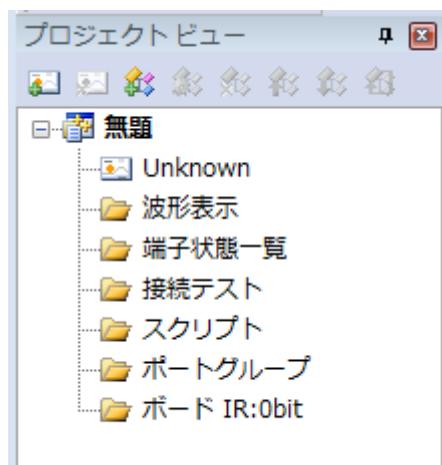


Figure 80

2) JTAG 信号の強制出力

ツールボタン	メニューバーの操作	ショートカットキー
---	プロジェクト → JTAG 信号の強制出力	---

上記メニューをクリックしてチェックを付けた状態にします。
 チェックを付ける際に確認のメッセージが表示されるので"OK"を選択してください。
 チェックを付けることで JTAG 信号の High レベルとして 3.3V を出力するようになります。

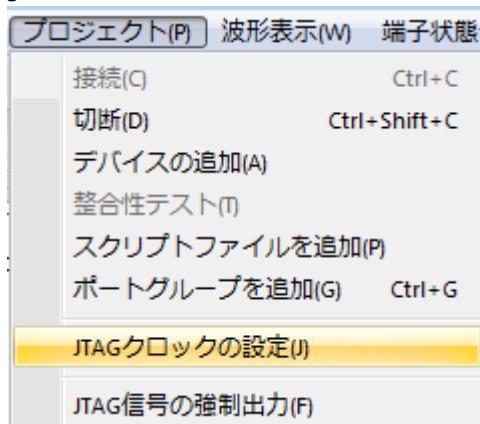


Figure 81

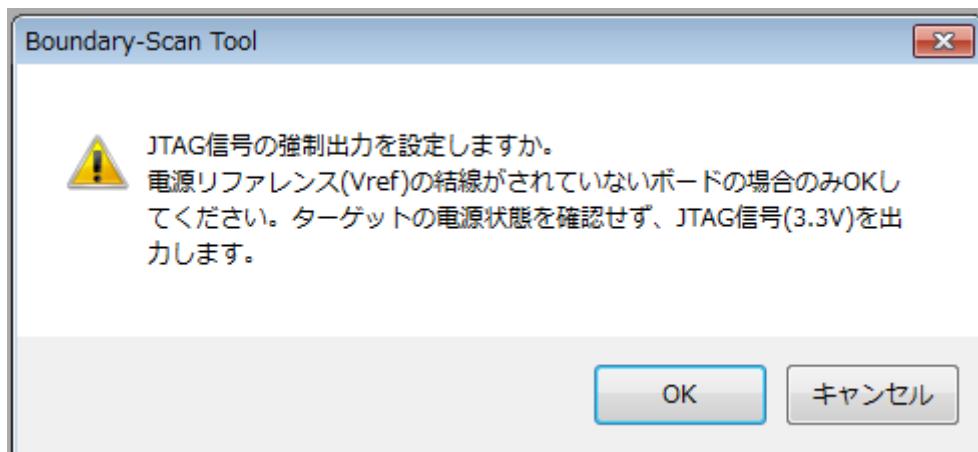


Figure 82

3) 接続

ツールボタン	メニューバーの操作	ショートカットキー
	プロジェクト → 接続	Ctrl + C

本ソフトウェアと、プローブを接続します。接続時にターゲットのバウンダリスキャンチェーンの自動認識を試みます。

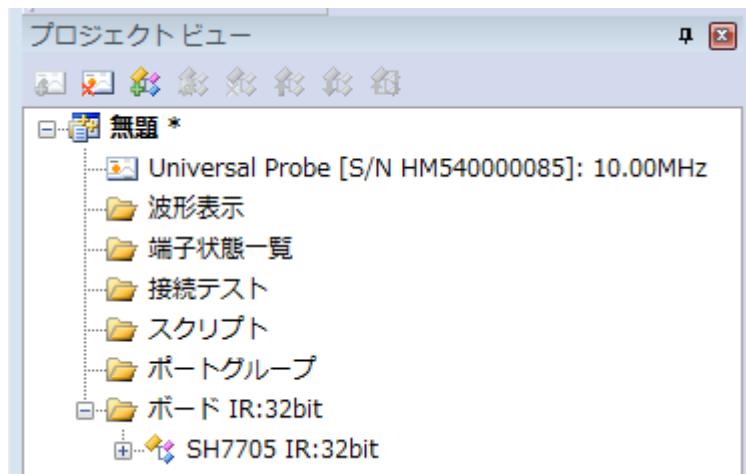


Figure 83

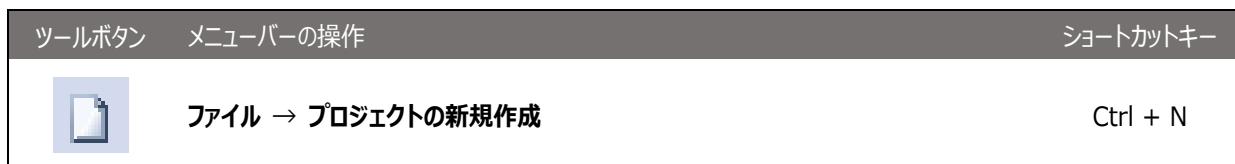


A.1.2. BSDL 情報がないデバイスが認識される場合

ハードウェアとしては一つの CPU ですが、バウンダリスキャンチェーンの自動認識をさせると複数のデバイスが検出されることがあります。そのような場合は以下手順でバウンダリスキャンテストを実行できる可能性があります（例：i.MX512 など）。

*i.MX512 は Freescale Corporation の商標です。

1) プロジェクトの新規作成



上記メニューを実行すると、プロジェクトビューに「無題」プロジェクトが作成されます。

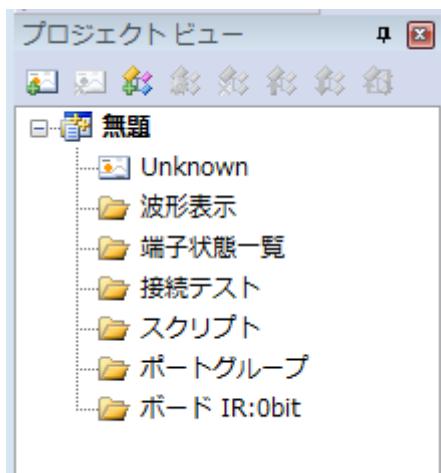


Figure 84

2) 接続



本ソフトウェアと、プローブを接続します。接続時にターゲットのバウンダリスキャンチェーンの自動認識を試みます。

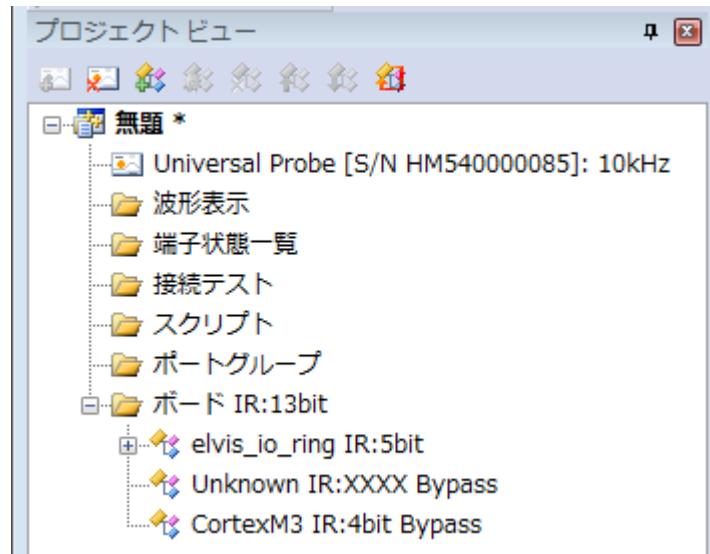


Figure 85

3) "unknown"デバイスのプロパティ

ツールボタン	メニューバーの操作	ショートカットキー
---	メニューbaruからの操作はできません。	
---	*プロジェクトビュー → ボード → unknown デバイス 右クリックによる コンテキストメニュー	---

選択した"unknown"デバイスのプロパティを表示します。

プロジェクトビューの"ボード"欄にそのボードの総 IR ビット長が表示されていますので、他のデバイスの IR ビット長と比較して総 IR ビット長に合うようにデバイスプロパティダイアログの IR 長を編集してください。

また、"Bypass 状態にする"にチェックを付けてください。"unknown"デバイスはポートの情報がないので、バウンダリスキヤンテスト時に BYPASS 命令によりテストをスキップするためです。

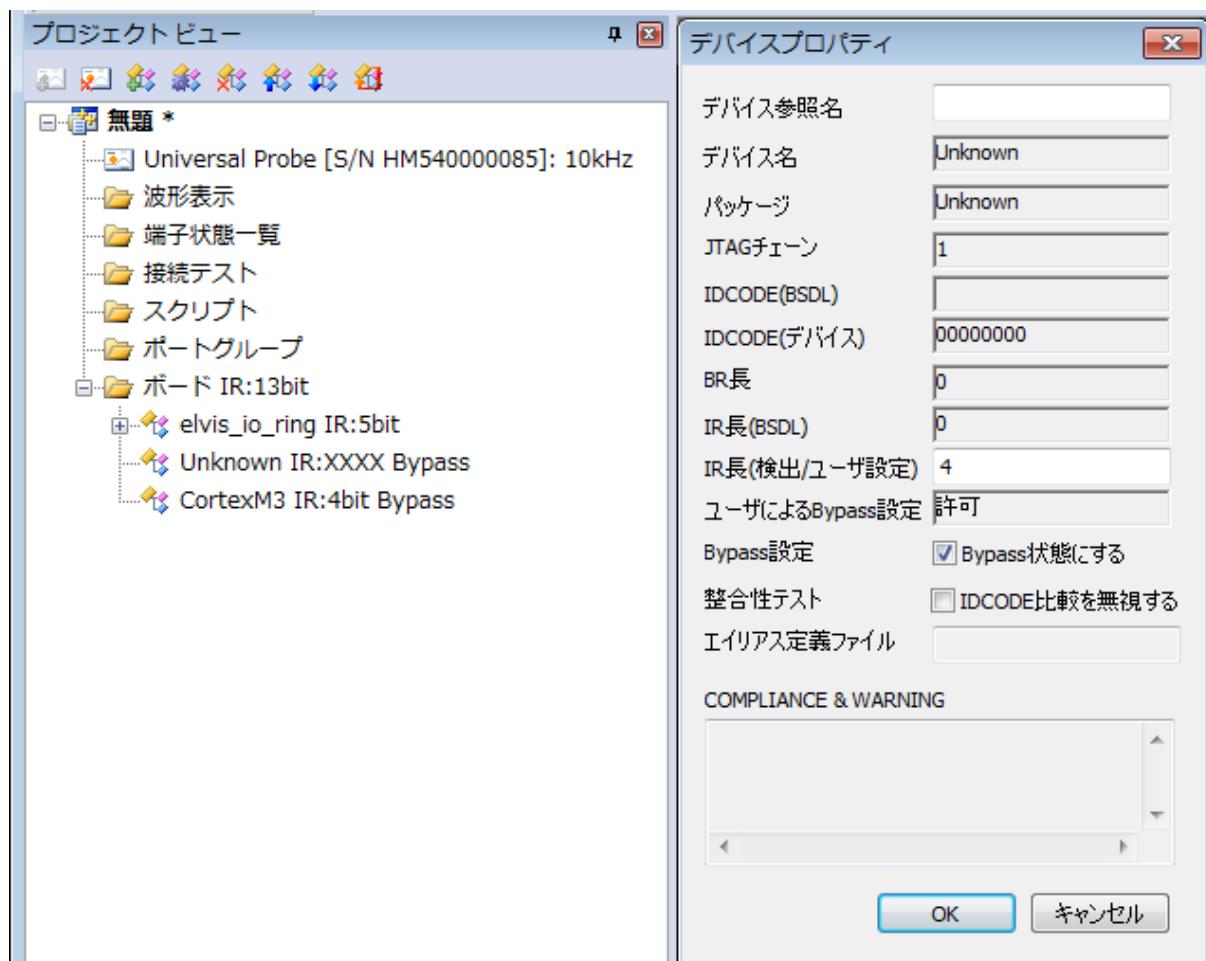


Figure 86

4) "IDCODE の無視"設定

BSDLに記述されているIDCODE情報とデバイスから実際に取得されたIDCODEが食い違っているデバイスは、デバイスプロパティダイアログで"IDCODE 比較を無視する"にチェックを付けます。

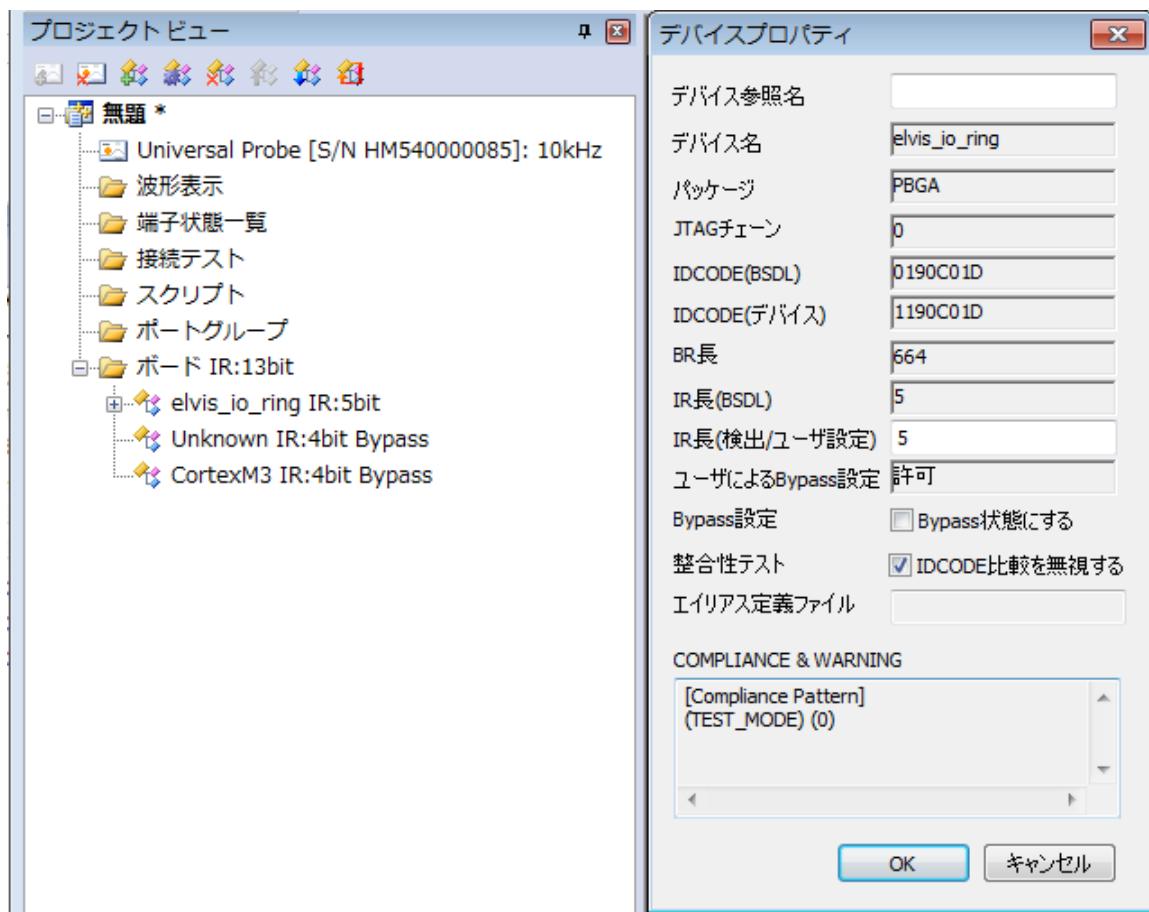
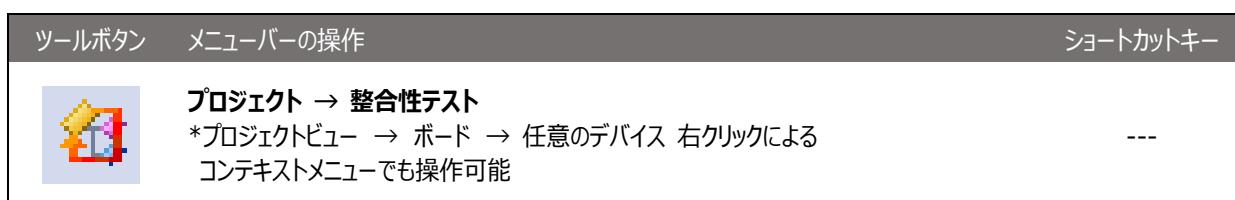


Figure 87

5) 整合性テスト



プロジェクトビューに構成されたバウンダリスキャンチェーンでバウンダリスキャンテストを行うことができるかをテストします。テストに成功すると上記メニューがグレーアウト状態になります。

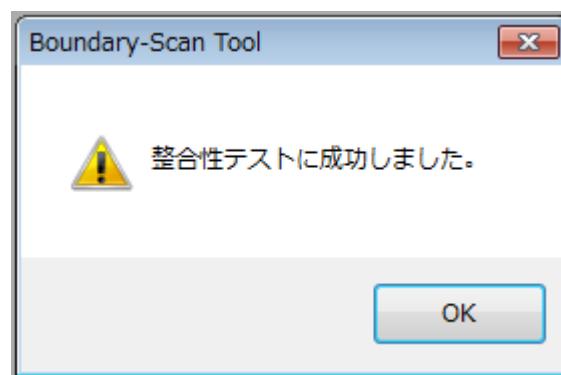


Figure 88

A.2. メッセージ一覧

A.2.1. [ERROR]

エラーメッセージとその説明を以下に示します。

Table 43

1	「プローブが検出できませんでした。 USB ポートを確認してください。」	"接続"メニュー実行時に本ソフトウェアがプローブを検出できなかつた場合に表示されます。プローブがパソコンに接続されていないか、USB ケーブルもしくはプローブの故障が考えられます。
2	「指定したシリアル番号のプローブが検出 できませんでした。他のプローブと接続しま すか？」	"接続"メニュー実行時に本ソフトウェアがプロジェクトビューに登録されているプローブを検出できなかつた場合に表示されます。 プローブがパソコンに接続されていない、 プロジェクトビューに登録しているものと違うプローブがパソコンに接続されている、USB ケーブルもしくはプローブの故障が考えられます。
3	「プロジェクトファイルが開けませんでした。」	"プロジェクトを開く"メニュー実行時にファイルアクセスに失敗したなどの理由でプロジェクトファイルの内容をプロジェクトに復元できなかつた際に表示されます。
4	「BSDL ファイルからのポート取得に失敗し ました。」	"プロジェクトを開く"及び"デバイス編集"メニュー実行時に、 指定した BSDL ファイルの BSDL 情報からポート群を作成できなかつたときに表示されます。
5	「BSDL ファイルの解析に失敗しました。」	"デバイス編集ダイアログ"及び"BSDL 管理ダイアログ"で、 BSDL ファイルを読み込んだ後、ファイルの解析に失敗した場合に表示されます。
6	「SAMPLE コマンドの実行に失敗しまし た。」	端子状態一覧ウィンドウの"SAMPLE コマンド"メニュー実行に失敗すると表示されます。
7	「EXTEST コマンドの実行に失敗しまし た。」	端子状態一覧ウィンドウの"EXTEST コマンド"メニュー実行に失敗すると表示されます。
8	「IR の長さが設定されていません。」	端子状態一覧ウィンドウの"SAMPLE コマンド"及び "EXTEST コマンド"メニュー実行時に IR ビット長が"0"の場合に表示されます。
9	「BSDL が求まっていないデバイスは Bypass 設定してください。」	端子状態一覧ウィンドウの"SAMPLE コマンド"及び "EXTEST コマンド"メニュー実行時に表示されます。 プロジェクトビューで"Unknown"となっているデバイスがバイパスに設定されていないときに表示されます。
10	「全体の IR の長さが設定と一致しませ ん。」	端子状態一覧ウィンドウの"SAMPLE コマンド"及び "EXTEST コマンド"メニュー実行時に表示されます。プロジェクトビューで各デバイスに設定された IR ビット長とターゲットの各デバイスから読み出せた IR ビット長が異なる場合に表示されます。
11	「製品 ID 取得に失敗しました。インストー ルフルオルダの wpcpu.id が存在しないか壊 れている可能性があります。」	"接続"メニュー実行時に左記のファイルから製品 ID を取得できなかつた場合に表示されます。
12	「ライセンスを確認できませんでした。 License.No と プローブのシリアル番号 の組み合わせをご確認ください。」	"接続"及び"プローブのライセンス取得"メニュー実行でライセンス取得画面を表示した後、ライセンス取得に失敗した場合に表示されます。
13	「サンプリングでエラーが発生しました。停 止します。」	波形表示ウィンドウで計測中にエラーが発生すると表示されます。
14	「波形ファイル「ファイル名」のフォーマットが 不正です。」	波形を"ビューワで開く"メニュー実行時に不正なフォーマットのファイルを開くと表示されます。

A.2.2. [WARNING]

ワーニングメッセージとその説明を以下に示します。

Table 44

1 「検出デバイス数 n Device 1 : IDCODE Device 2 : IDCODE ...」	"接続"メニュー実行時にデバイスを自動認識した場合に表示されます。
2 「プローブを切斷します。プロジェクトビューのプローブ情報をクリアしますか？」	"切斷"メニュー実行時にプローブのシリアル及び JTAG 周波数の情報をプロジェクトに残すか問い合わせてきます。
3 「サンプリングを停止します。よろしいですか？」	"プロジェクトの新規作成"、"プロジェクトを開く"、"プロジェクトを閉じる"、"アプリケーションの終了"、"波形表示パターンの新規作成"、"波形の保存"の各メニューを実行する際に波形表示ウィンドウで計測中なら問い合わせてきます。
4 「プロジェクトに登録されているプローブを検出しました。接続しますか？」	"プロジェクトを開く"メニュー実行時に、プロジェクトファイルに登録されているプローブが パソコンに接続されれば問い合わせてきます。
5 「現在のプロジェクトは変更されています。保存しますか？」	プロジェクトビューにプロジェクトがあり、プロジェクト名に "*" が付いている状態で、"プロジェクトの新規作成"、"プロジェクトを開く"、"プロジェクトを閉じる"、"アプリケーションの終了" の各メニューを実行すると問い合わせてきます。
6 「以下のデバイスを登録しました。 メーカー名: デバイス 1 デバイス 2...」	"デバイス編集ダイアログ" 及び "BSDL 管理ダイアログ" で BSDL ファイルを読み込んだ後、ファイルの解析に成功した場合に表示されます。
7 「整合性テストに成功しました。」	"整合性テスト"メニューの実行結果を表示します。
8 「整合性テストに失敗しました。」	"整合性テスト"メニューの実行結果を表示します。
9 「選択したリストは端子状態一覧ウィンドウで使用中です。左記のウィンドウを閉じるか別のリストに切り替えてから削除してください。」	端子状態一覧ウィンドウで開いているリストをプロジェクトビューから削除しようとすると表示されます。
10 「選択したパターンは波形表示ウィンドウで使用中です。左記のウィンドウを閉じるか別のパターンに切り替えてから削除してください。」	波形表示ウィンドウで開いているパターンをプロジェクトビューから削除しようとすると表示されます。
11 「整合性テストを通過済みですが処理を続けますか？」	整合性テストをパスした状態（バウンダリスキヤンテストを実行可能な状態）で、"デバイスの追加"、"デバイスの編集"、"デバイスの上移動"、"デバイスの下移動" の各メニューを実行する際に問い合わせてきます。処理を継続すると整合性テストが通らなくなり、バウンダリスキヤンテストを実行できなくなる可能性があります。
12 「波形をクリアします。よろしいですか？」	波形表示ウィンドウで、"信号追加"、"信号削除"、"計測開始"、"サンプリング間隔の変更"、の各メニューを実行するときにウィンドウに波形データが表示された場合に問い合わせてきます。
13 「実行中のスクリプトが入力待ちです。アプリケーションの終了はできません。」	スクリプトペインで、Python スクリプトの input() 命令などで入力待ちになっているときに、アプリケーションの終了メニューを実行しようとすると表示されます。スクリプトの実行を完了させてからアプリケーションを終了してください。
14 「"グループ名" は削除されたデバイスのポートを含んでいたため削除します。」	プロジェクトビューでデバイスを削除した際にそのデバイスが持っているポートがグループに登録されていると表示されます。

15 「このスクリプトファイルは登録済みです。」	プロジェクトメニューの『スクリプトファイルの追加』やプロジェクトビューでスクリプトファイルを選択したときに、そのファイルが既に登録済みであればこのメッセージが表示されます。
16 「指定されたスクリプトファイルが見つかりませんでした。」	スクリプトファイルの実行時に指定されたファイルパスにファイルが存在しない場合にこのメッセージが表示されます。

A.3. トラブルシューティング

A.3.1. インストール関係

質問	回答
プローブが検出できない。	<p>下記を確認して下さい。</p> <p>① 本ソフトウェアは Universal Probe 以外のプローブは検出しません。 ② プローブとパソコンが適切に接続されているか再度ご確認ください。 『電源の ON-OFF 手順』も併せてご参照ください。 ③ ソフトウェアのインストールが失敗して、ドライバソフトウェアが インストールされていない可能性があります。 再度、インストールしてみて下さい。</p>



申し訳ありませんが、パソコン側の仕様により、接続できない場合が存在します。
 どうしても、接続できない場合は、パソコンを変更してみて下さい。

A.3.2. プロジェクトビュー使用時

質問	回答
デバイスの自動検出に失敗する。	<p>下記を確認して下さい。</p> <p>① ボードの電源及び JTAG ケーブルの接続を確認してください。 JTAG ケーブルの電源リファレンス(Vref)信号が結線されていないボードの場合は、 "プロジェクト >> JTAG 信号の強制出力"メニューにチェックを付けることで認識できる 可能性があります。</p> <p>② ツールメニューから "BSDL 管理" ダイアログを開いて目的のデバイスが登録されているか 確認してください。 登録されていない場合はデバイスマーカーの WEB サイト等から BSDL ファイルを入手し、 上記ダイアログからデバイスの登録を行ってください。 詳細については『ツール』の "BSDL 管理" をご参照ください。</p> <p>BSDL ファイルが公開されていない場合は、申し訳ございませんがそのデバイスのバウンダリ スキヤンテストを行うことはできません。</p> <p>③ "②"で目的のデバイスが登録されている場合は、"BSDL 管理" ダイアログで目的のデバ イスの IDCODE を確認してください。 IDCODE 欄が空であれば、『デバイスの追加』でデバイスを登録し、 プロジェクトビューの"デバイスのプロパティ"で"IDCODE 比較を無視する"にチェックを付 けることで バウンダリスキヤンテストを実行できる可能性があります。</p> <p>④ "③"で解決しない場合は同じく"デバイスのプロパティ"で"IR ビット長"を変更するこ とで バウンダリスキヤンテストを実行できる可能性があります。</p> <ul style="list-style-type: none"> これは BSDL ファイルの IR ビット長定義とデバイスの IR ビット長が異なる場合の み有効です。 <p>⑤ 上記全ての場合で解決しない場合はお手数ですが弊社へお問い合わせください。</p>

改訂履歴

版数	改訂日	改訂内容
01	2014/09/30	初版
02	2014/11/26	誤記修正

製造者情報



株式会社 Sohwa & Sophia Technologies

〒215-8588

[本社]

神奈川県川崎市麻生区南黒川 6-2

ホームページ : <http://www.ss-technologies.co.jp>

子会社



SOHWA MALAYSIA SND. BHD.

Unit 5-2, Level 5, Tower 6, Avenue 5,
The Horizon, Bangsar South No.8, Jalan Kerinchi
59200, Kuala Lumpur, Malaysia

HomePage : <http://www.sohwa-m.com.my/>