

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

ユーザーズ・マニュアル

Applilet[®]2

デバイス・ドライバ・コンフィギュレータ

共通操作編

78K0 マイクロコントローラ

78K0R マイクロコントローラ

V850 マイクロコントローラ

[メ モ]

Applilet, All Flash および MINICUBE は, NEC エレクトロニクス株式会社の日本およびその他の国における登録商標または商標です。

IAR Embedded Workbench は, IAR Systems AB の登録商標または商標です。

MULTI は, 米国 Green Hill Software, Inc. の米国およびその他の国における登録商標または商標です。

Pentium は, 米国 Intel Corporation の米国およびその他の国における登録商標または商標です。

Windows は, 米国 Microsoft Corporation の米国, 日本およびその他の国における登録商標または商標です。

その他, この資料に記載されている会社名, 製品名などは, 各社の商標または登録商標です。

- 本資料に記載されている内容は2008年12月現在のもので, 今後, 予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は, 本資料の誤りに関し, 一切その責を負いません。
- 当社は, 本資料に記載された当社製品の使用に関連し発生した第三者の特許権, 著作権その他の知的財産権の侵害等に関し, 一切その責を負いません。当社は, 本資料に基づき当社または第三者の特許権, 著作権その他の知的財産権を何ら許諾するものではありません。
- 本資料に記載された回路, ソフトウェアおよびこれらに関連する情報は, 半導体製品の動作例, 応用例を説明するものです。お客様の機器の設計において, 回路, ソフトウェアおよびこれらに関連する情報を使用する場合には, お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し, 当社は, 一切その責を負いません。
- 当社は, 当社製品の品質, 信頼性の向上に努めておりますが, 当社製品の不具合が完全に発生しないことを保証するものではありません。また, 当社製品は耐放射線設計については行っておりません。当社製品をお客様の機器にご使用の際には, 当社製品の不具合の結果として, 生命, 身体および財産に対する損害や社会的損害を生じさせないよう, お客様の責任において冗長設計, 延焼対策設計, 誤動作防止設計等の安全設計を行ってください。
- 当社は, 当社製品の品質水準を「標準水準」, 「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また, 各品質水準は, 以下に示す用途に製品が使われることを意図しておりますので, 当社製品の品質水準をご確認ください。

標準水準: コンピュータ, OA機器, 通信機器, 計測機器, AV機器, 家電, 工作機械, パーソナル機器, 産業用ロボット

特別水準: 輸送機器(自動車, 電車, 船舶等), 交通用信号機器, 防災・防犯装置, 各種安全装置, 生命維持を目的として設計されていない医療機器

特定水準: 航空機器, 航空宇宙機器, 海底中継機器, 原子力制御システム, 生命維持のための医療機器, 生命維持のための装置またはシステム等

当社製品のデータ・シート, データ・ブック等の資料で特に品質水準の表示がない場合は, 標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には, 事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは, NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは, (1)において定義された当社の開発, 製造製品をいう。

M8E0710J

はじめに

対象者 このマニュアルは、78K0 マイクロコントローラ、78K0R マイクロコントローラ、および V850 マイクロコントローラ用デバイス・ドライバ・コンフィギュレータ Applilet2 の機能を理解し、それをを用いたアプリケーション・システムを設計するユーザを対象とします。

目的 このマニュアルは、Applilet2 の持つソフトウェア機能をユーザに理解していただき、これを使用するシステムのハードウェア、ソフトウェア開発の参照用資料として役立つことを目的としています。

構成 このマニュアルは、大きく分けて次の内容で構成しています。

- 第 1 章 概 説
- 第 2 章 インストレーション
- 第 3 章 操作方法
- 第 4 章 メニュー・リファレンス
- 第 5 章 ウィンドウ・リファレンス
- 第 6 章 ダイアログ・リファレンス
- 第 7 章 出力ファイル

読み方 このマニュアルを読むにあたっては、電気、論理回路、マイクロコンピュータに関する一般的知識が必要となります。

- ・ Applilet2 の機能の詳細を理解しようとするとき
目次に従ってお読みください。

凡 例	[]	: メニュー名, ボタン名, タブ名
	「 」	: ダイアログ名, ウィンドウ名
	注	: 本文中につけた注の説明
	注意	: 気をつけて読んでいただきたい内容
	備考	: 本文中の補足説明
	数の表記	: 2 進数 ... x x x x または x x x x b
		10 進数 ... x x x x
		16 進数 ... x x x x H または 0x x x x x

用語説明 このマニュアルで使用する用語を次の表に示します。

用語	意味
NEC 環境	NEC エレクトロニクス社製の言語ツールおよび統合開発環境プラットフォームを使用してプログラム開発を行う環境
IAR 環境	IAR システムズ社製の言語ツールおよび統合開発環境プラットフォームを使用してプログラム開発を行う環境
GHS 環境	Green Hills Software 社製の言語ツールおよび統合開発環境プラットフォームを使用してプログラム開発を行う環境
78K0 系 Applilet2	78K0S および 78K0 マイクロコントローラ用の Applilet2
78K0R 系 Applilet2	78K0R マイクロコントローラ用の Applilet2
V850 系 Applilet2	V850 マイクロコントローラ用の Applilet2

備考 Applilet2 は、製品ごとに GUI デザインが異なります。

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

資料名	資料番号	
CC78K0 Ver.3.70 ユーザーズ・マニュアル	操作編	U17201J
	言語編	U17200J
CC78K0R Ver.2.00 ユーザーズ・マニュアル	操作編	U18548J
	言語編	U18549J
CA850 Ver.3.20 ユーザーズ・マニュアル	操作編	U18512J
	言語編	U18513J
RA78K0 Ver.3.70 ユーザーズ・マニュアル	操作編	U17199J
	言語編	U17198J
RA78K0R Ver.1.20 ユーザーズ・マニュアル	操作編	U18546J
	言語編	U18547J
PM plus Ver.5.20 ユーザーズ・マニュアル		U16934J
PM+ Ver.6.30 ユーザーズ・マニュアル		U18416J
QB-MINI2 ^注 ユーザーズ・マニュアル		U18371J
QB-MINI2 ^注 セットアップ・マニュアル	パートナー・ツール編	U19158J

注 QB-MINI2：プログラミング機能付きオンチップ・デバッグ・エミュレータ MINICUBE[®]2

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料を使用してください。

目 次

第1章 概 説 ... 9

- 1.1 概 要 ... 9
- 1.2 開発フロー ... 9
- 1.3 機 能 ... 10
- 1.4 動作環境 ... 11

第2章 インストレーション ... 12

- 2.1 インストーラの特徴 ... 12
- 2.2 フォルダ構成 ... 12
- 2.3 インストール手順 ... 13
- 2.4 アンインストール手順 ... 15

第3章 操作方法 ... 17

- 3.1 各部の名称 ... 17
 - 3.1.1 タイトル・バー ... 18
 - 3.1.2 メニュー・バー ... 18
 - 3.1.3 ツール・バー ... 18
 - 3.1.4 ツリー・ビュー・ウインドウ ... 19
 - 3.1.5 情報表示ウインドウ ... 19
 - 3.1.6 ステータス・バー ... 20
- 3.2 操作手順 ... 21
- 3.3 起 動 ... 22
- 3.4 新規作成 ... 23
- 3.5 既存のプロジェクトを開く ... 25
- 3.6 周辺機能の設定 ... 26
- 3.7 コード生成 ... 27
 - 3.7.1 コード生成先フォルダを変更する ... 28
 - 3.7.2 上書き確認 ... 30
 - 3.7.3 マージ機能 ... 31
- 3.8 プロジェクトを保存する ... 33
- 3.9 終 了 ... 34
- 3.10 コーディング ... 34

第4章 メニュー・リファレンス ... 35

- 4.1 [ファイル(F)] メニュー ... 35

- 4.2 [周辺機能(P)] メニュー ... 36
- 4.3 [オプション(O)] メニュー ... 37
- 4.4 [ヘルプ(H)] メニュー ... 38
- 4.5 ツール・バー ... 39
 - 4.5.1 ツール・バー1 ... 39
 - 4.5.2 ツール・バー2 ... 40

第5章 ウィンドウ・リファレンス ... 41

- 5.1 [周辺機能] タブ ... 41
- 5.2 [メモリ] タブ ... 43
- 5.3 [関数] タブ ... 44
- 5.4 [端子] タブ ... 46
- 5.5 [I/O レジスタ] タブ ... 48
- 5.6 [割り込み] タブ ... 50

第6章 ダイアログ・リファレンス ... 52

- 6.1 周辺機能設定ダイアログ ... 52
 - 6.1.1 設定ダイアログの構成 ... 52
 - 6.1.2 設定方法 ... 53
 - 6.1.3 設定ダイアログ例 ... 57
- 6.2 「詳細情報」ダイアログ ... 64
 - 6.2.1 [I/O レジスタ] タブ ... 64
 - 6.2.2 [関数] タブ ... 65
 - 6.2.3 [割り込み] タブ ... 66
 - 6.2.4 [端子] タブ ... 67
- 6.3 「コード生成」ダイアログ ... 68
- 6.4 「設定情報出力」ダイアログ ... 70

第7章 出力ファイル ... 71

- 7.1 V850系 Applilet2 ... 71
 - 7.1.1 NEC 環境での出力ファイル ... 71
 - 7.1.2 IAR 環境での出力ファイル ... 72
 - 7.1.3 GHS 環境での出力ファイル ... 73
- 7.2 78K0R系 Applilet2 ... 74
 - 7.2.1 NEC 環境 C 言語ソースでの出力ファイル ... 74
 - 7.2.2 NEC 環境 ASM 言語ソースでの出力ファイル ... 75
 - 7.2.3 IAR 環境 C 言語ソースでの出力ファイル ... 76
 - 7.2.4 IAR 環境 ASM 言語ソースでの出力ファイル ... 77
- 7.3 78K0系 Applilet2 ... 78
 - 7.3.1 NEC 環境 C 言語ソースでの出力ファイル ... 78
 - 7.3.2 NEC 環境 ASM 言語ソースでの出力ファイル ... 79

7.3.3	IAR 環境 C 言語ソースでの出力ファイル ...	80
7.3.4	IAR 環境 ASM 言語ソースでの出力ファイル ...	81

第1章 概 説

1.1 概 要

Applilet2 は NEC エレクトロニクス製 78K0 マイクロコントローラ，78K0R マイクロコントローラ，および V850 マイクロコントローラ用のデバイス・ドライバを自動生成するソフトウェア・ツールです。使用するデバイスと対応する Applilet2 を使用してください。

周辺機能モジュール（ペリフェラル）レジスタの初期設定を，GUI を通じて簡単に行うことができます。

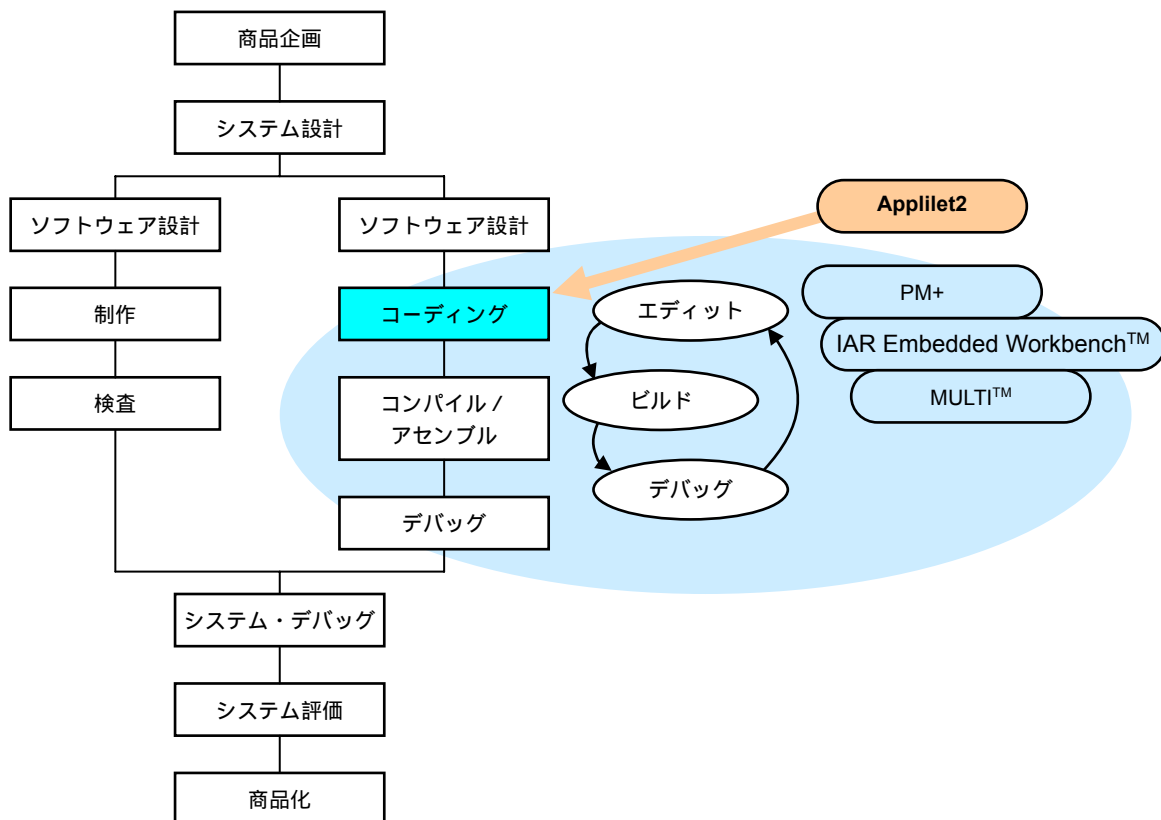
本マニュアルでは，Applilet2 のメイン・ウインドウ，メニューおよびダイアログの使用方法など，対象デバイスに依存しない部分の共通操作仕様について説明します。

本マニュアルでは，78K0R/KX3 用 Applilet2 の画面を例にして説明しています。

1.2 開発フロー

Applilet2 を使用した開発フローを次に示します。

図 1-1 開発フロー



備考 PM+：プロジェクト・マネージャ

IAR Embedded Workbench：IAR システムズ社製の統合開発環境

MULTI：Green Hills Software 社製の統合開発環境

1.3 機 能

デバイス・ドライバの出力

GUI 上で設定された内容に従って、周辺機能の初期化を行うソース・コードを自動生成し、ファイルとして出力します。ファイル名は任意に変更できます。

API の提供

周辺機能の初期化コードのほかに、周辺機能の動作 / 停止、条件変更などの API を提供します。

API 名は任意に変更できます。

ソース言語の選択

出力するソースの言語を C コンパイラ / アセンブラから選択できます (78K0S, 78K0, 78K0R のみ)。

言語ツールの選択

言語ツールの種類を NEC, IAR, GHS から選択できます (GHS は, V850 のみ)。

選択した言語ツールに合わせた、統合開発環境プラットフォームのワーク・スペース / プロジェクト・ファイルを出力します。

- NEC 環境 : PM+用のワーク・スペース・ファイルを出力 (.prw)
- IAR 環境 : IAR Embedded Workbench 用ワーク・スペース・ファイルを出力 (.eww)
- GHS 環境 : MULTI 用プロジェクト・ファイルを出力 (.gpj)

マージ機能

• ソース・コードのマージ

マージ用コメント間に書いたプログラムは、コード再出力 (上書き) 時、削除せずに残すことができます。

• ワーク・スペース / プロジェクト・ファイルのマージ

Applilet2 は、出力ファイルをビルド対象のファイルとして、統合開発環境プラットフォームのワーク・スペース / プロジェクト・ファイルに登録します。再コード生成時、Applilet2 が出力するファイルの増減^注に合わせて、ビルド対象のファイルの登録を変更します。このとき、すでに登録されているユーザ・ファイルを削除せずに残します。

注 IAR, GHS 環境のとき、Applilet2 はファイルの追加登録を行いますが、不要になったファイルの削除を行いません。

設定情報出力機能

周辺機能の情報や Applilet2 上で行った設定状況をファイルに出力できます。

- メモリ・マップ情報
- 周辺機能情報
- 端子情報
- 割り込み情報
- I/O, SFR レジスタ情報

1.4 動作環境

Applilet2 を使用する際は、次の条件を満たすホスト・マシンを使用してください。

表 1 - 1 動作環境

項 目	製品 / 性能
対応 OS ^{注1}	Windows® 2000 / Windows XP / Windows Vista
CPU	Pentium® III 500 MHz 以上
メモリ	512 MB 以上
その他	.NET Framework Version 2.0 ^{注2} のランタイムおよび関連ファイルがインストールされた環境で動作します。

- 注 1. 対応する環境にインストールしても、オペレーティング・システムのローカル・バージョンの違いによっては動作しない場合があります。
起動しない場合は、オペレーティング・システムのアップデートやサービス・パックを適用してください。
2. “.NET Framework Version 2.0” は、最新のアップデート・プログラムを適用した状態で使用してください。

第2章 インストール

2.1 インストーラの特徴

Applilet2 インストーラの特徴を次に示します。

複数バージョン対応

1つのPCに複数のバージョンをインストールできます。

日英環境共通パッケージ

インストーラは言語の自動判別を行いません。インストール開始時に、ユーザが言語を選択します。

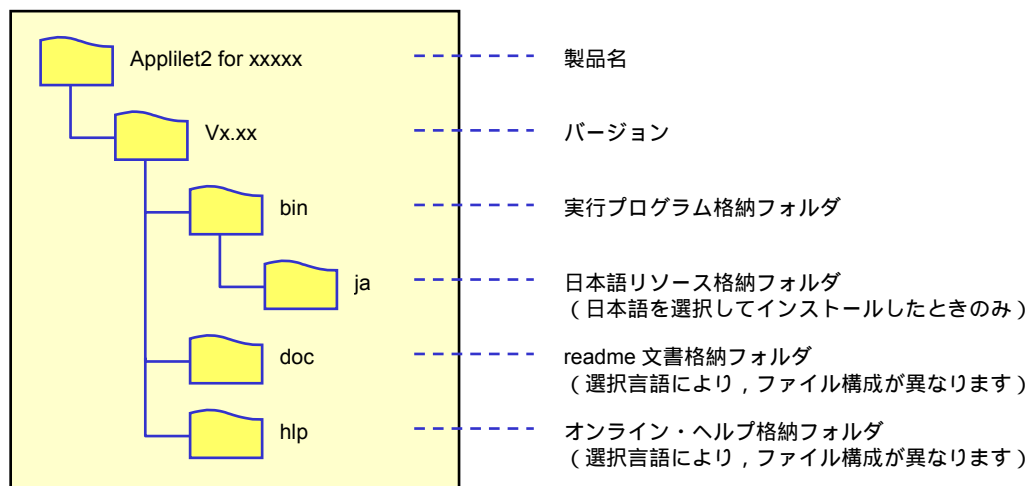
日本語版 Windows 上でも「英語」を選択してインストールした場合は、英語表示の Applilet2 をインストールできます。

注意 Applilet2 は単独のアプリケーションとして使用します。PM+との連携に必要なレジストリへの書き込みを行いません。

2.2 フォルダ構成

Applilet2 のフォルダ構成は、次のとおりです。

図 2-1 フォルダ構成



2.3 インストール手順

Applilet2 をインストールする手順を次に示します。

Windows XP に Applilet2 for 78K0RKX3 をインストールする場合を例として示します。使用しているオペレーティング・システムおよびソフトウェアによって表示される内容が異なります。

- 注意**
- Administrator 権限のあるユーザでログインしてインストールする必要があります。
 - Applilet2 の実行には “.NET Framework Version 2.0 ” のランタイムと関連ファイルをインストールする必要があります。使用するホスト・マシンにこれらのファイルがインストールされていない場合、Microsoft 社のホームページからダウンロードしてインストールしてください。インストールされているかどうかは、Windows の「プログラムの追加と削除」で確認できます。

図 2 - 2 プログラムの追加と削除 (.NET Framework Version 2.0 の確認)



(1) Applilet2 のインストーラの「Setup.exe」を実行します。

備考 1. Applilet2 のインストーラは、NEC エレクトロニクスのホームページから入手できます。

<http://www.necel.com/micro/ja/development/asia/applilet2/index.html>

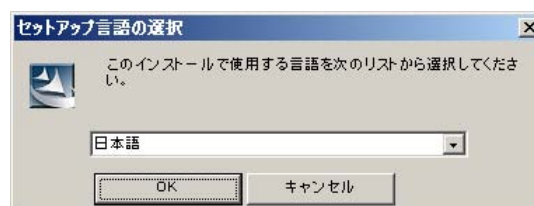
(ホームページのアドレスは、予告なく変更する場合があります。)

- ダウンロードしたインストーラは圧縮されている場合があります。圧縮されている場合、解凍してから「Setup.exe」を実行してください。

(2)「セットアップ言語の選択」ダイアログが表示されます。

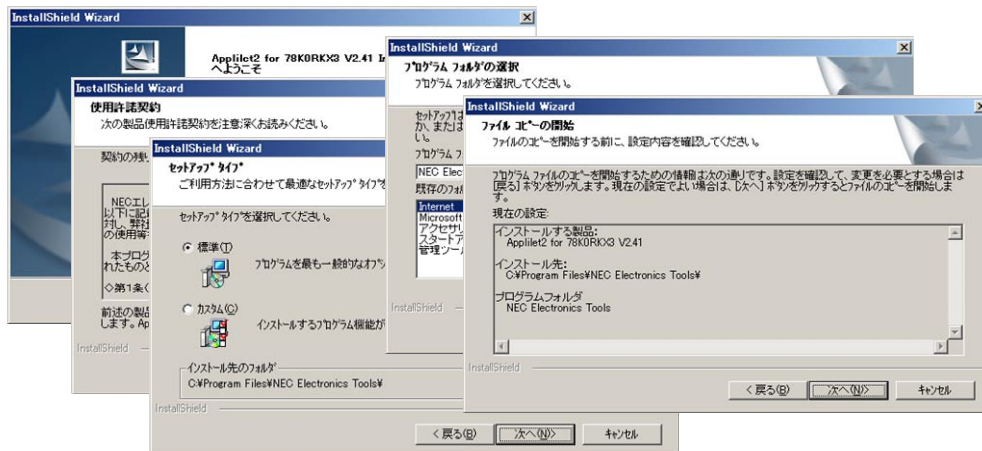
使用する言語を選択し、[OK]ボタンを押します。

図 2 - 3 「セットアップ言語の選択」ダイアログ



- (3) 表示されるウィザード・ダイアログの指示に従ってインストールの設定を行います。
各ダイアログでは、[次へ(N)] ボタンまたは [はい(Y)] ボタンを押すと次の画面に進みます。

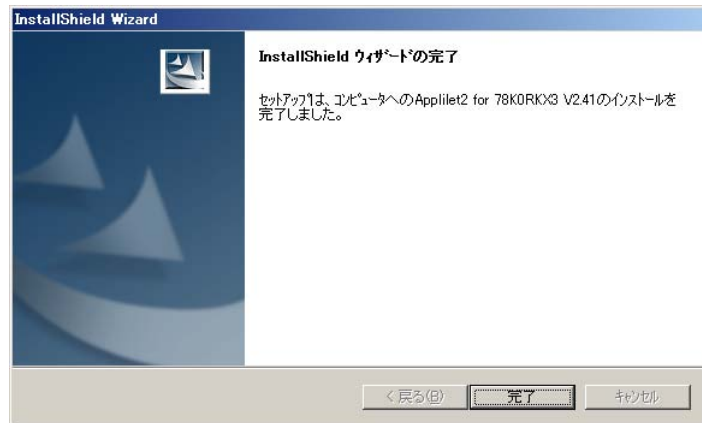
図 2-4 インストール・ウィザード・ダイアログ



注意 インストール先のフォルダ名には「(/ * : < > ? | ” ¥ ; , 」の 11 文字を使用できません。
また、スペース（半角の空白）をフォルダ名の先頭や末尾に使用することはできません。
フォルダ名に不正な文字を使用した場合、正常動作しません。

- (3) 「ウィザードの完了」画面で [完了] ボタンを押すとインストールを終了します。

図 2-5 「ウィザードの完了」画面



2.4 アンインストール手順

Applilet2 をアンインストールする手順を次に示します。

Windows XP で Applilet2 for 78K0RKX3 をアンインストールする場合を例として示します。使用しているオペレーティング・システムおよびソフトウェアによって表示される内容が異なります。

- 注意**
- Administrator 権限のあるユーザでログインしてアンインストールする必要があります。
 - Applilet2 をアンインストールしても “.NET Framework Version 2.0” のランタイムと関連ファイルはアンインストールされません。

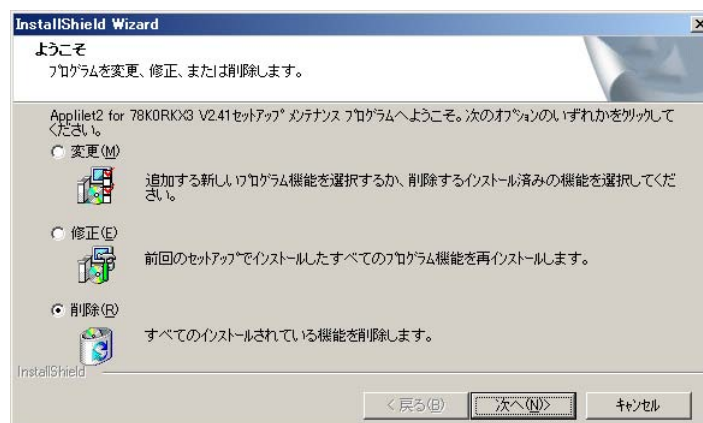
(1) Windows の「プログラムの追加と削除」からアンインストールする Applilet2 の [変更と削除] ボタンを押します。

図 2-6 プログラムの追加と削除 (Applilet2 の変更と削除)



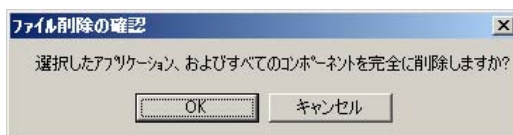
(2) 表示されるウィザード・ダイアログで [削除(R)] を選択し, [次へ(N)] ボタンを押します。

図 2-7 インストール・ウィザード (削除の選択)



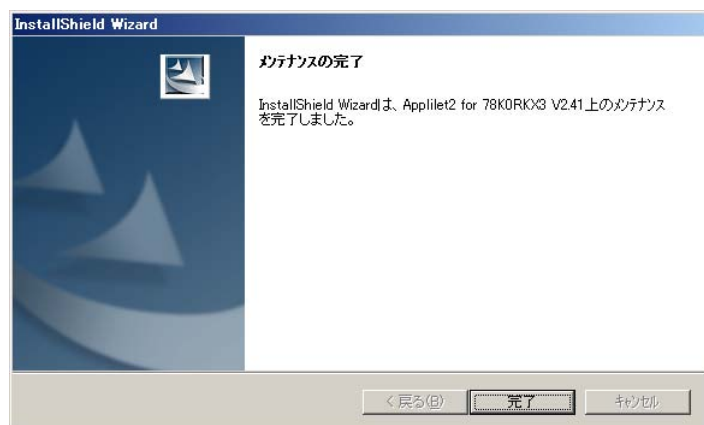
(3)「ファイル削除の確認」ダイアログが表示されたら，[OK] ボタンを押します。

図 2 - 8 「ファイル削除の確認」ダイアログ



(4)「メンテナンスの完了」画面で [完了] ボタンを押すとアンインストールを終了します。

図 2 - 9 「メンテナンスの完了」画面

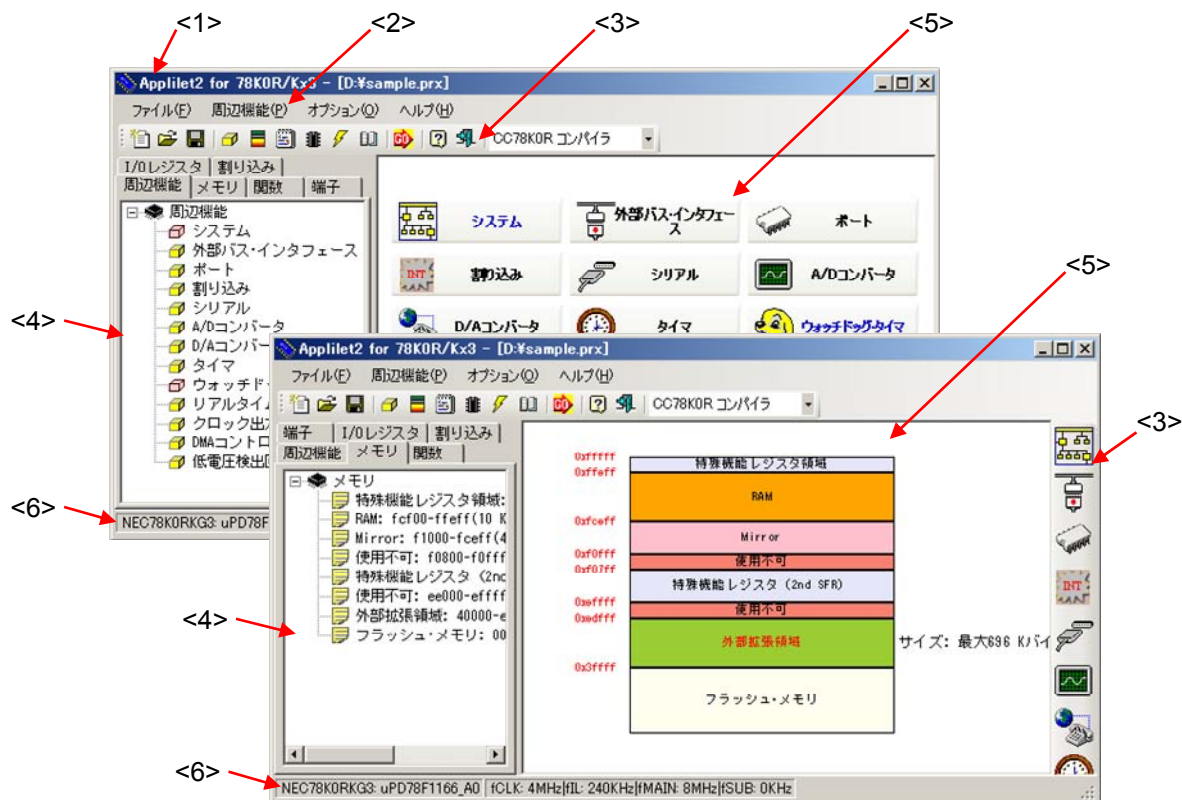


第3章 操作方法

3.1 各部の名称

Applilet2 の各部の名称を次に示します。

図 3-1 メイン・ウィンドウ



- <1> タイトル・バー : 製品名と Applilet2 のプロジェクト・ファイル名を表示します。
- <2> メニュー・バー : コマンドを選択して実行します。
- <3> ツール・バー : ボタンをクリックしてコマンドを選択して実行します。
- <4> ツリー・ビュー・ウィンドウ : タブで選択している項目の内容をツリー形式で表示します。
- <5> 情報表示ウィンドウ : ツリー・ビュー・ウィンドウで選択している項目に応じた内容を表示します。
- <6> ステータス・バー : 現在のプロジェクトの情報を表示します。

3.1.1 タイトル・バー

製品名と Applilet2 のプロジェクト・ファイル名を表示します。

プロジェクト・ファイル名に “*” が付いているときは、最新の設定内容が保存されていないことを示します。

図 3-2 タイトル・バー

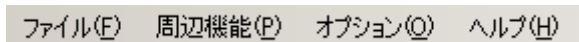


3.1.2 メニュー・バー

メニュー・バーからコマンドを選択して実行します。

各メニューの機能については、第4章 **メニュー・リファレンス**を参照してください。

図 3-3 メニュー・バー

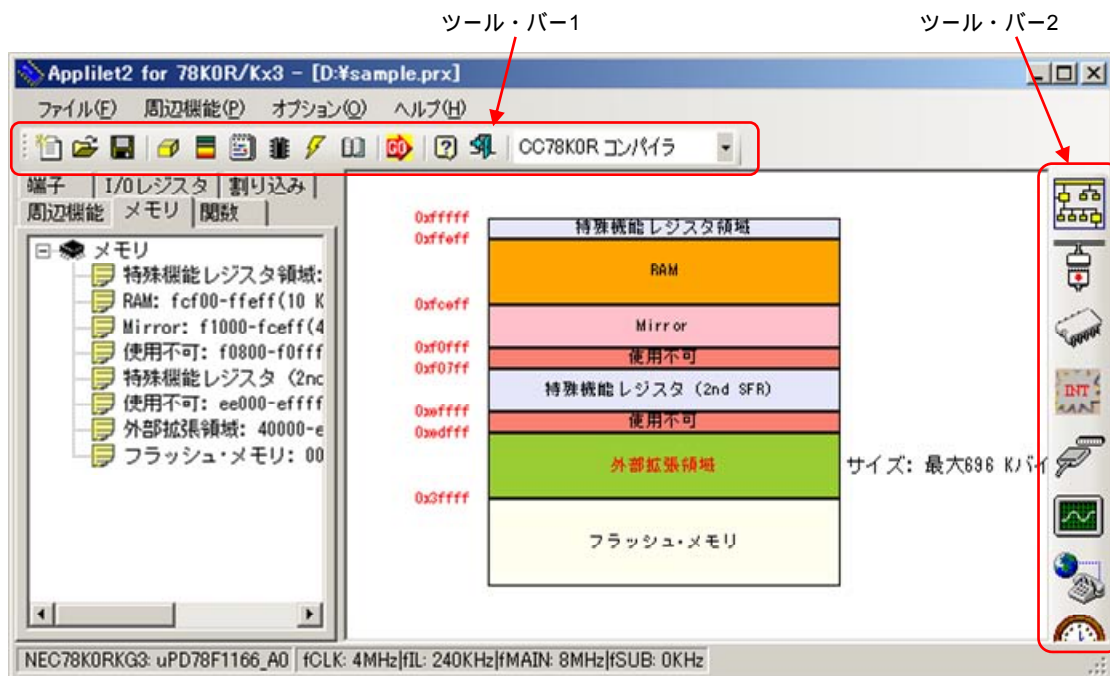


3.1.3 ツール・バー

ツール・バーは、メニュー・バーの下に常時表示しているツール・バー（ツール・バー1）と、ツリー・ビュー・ウィンドウにおいて、[周辺機能] タブ以外が選択されているときに、メイン・ウィンドウの右横に表示するツール・バー（ツール・バー2）があります。

各ボタンの機能については、第4章 **メニュー・リファレンス**を参照してください。

図 3-4 ツール・バー

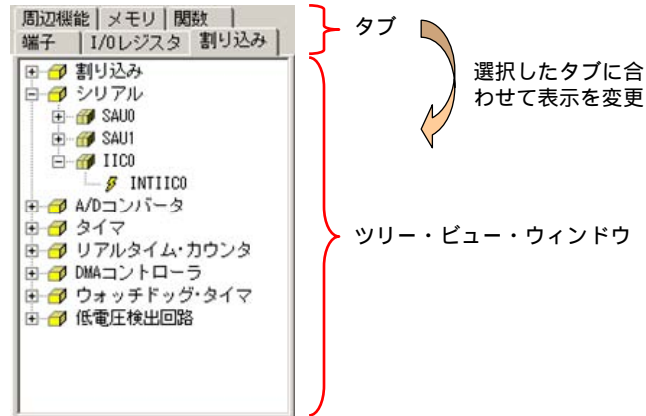


3.1.4 ツリー・ビュー・ウィンドウ

タブで選択している項目の内容をツリー形式で表示します。

表示内容については、第5章 ウィンドウ・リファレンスを参照してください。

図 3-5 ツリー・ビュー・ウィンドウとタブ



備考 ツリー・ビュー・ウィンドウの横幅が狭くすべての文字列が見えないときは、文字列またはアイコン上にマウス・カーソルを乗せると、ツールチップに各項目のすべての文字列を表示します。

3.1.5 情報表示ウィンドウ

ツリー・ビュー・ウィンドウで選択している項目に応じた内容を表示します。

表示内容については、第5章 ウィンドウ・リファレンスを参照してください。

ツリー・ビュー・ウィンドウで選択した項目に合わせて表示を変更



情報表示ウィンドウ

3.1.6 ステータス・バー

ステータス・バーには、現在のプロジェクトの情報を表示します。

図 3 - 6 ステータス・バー

(a) 起動直後、プロジェクトが読み込まれていない状態

プロジェクトを開くか新しいプロジェクトを作成してください

(b) プロジェクトを読み込んでいる状態 (78K0 系 Applilet2 , 78K0R 系 Applilet2)

NEC78K0KF2: uPD78F0547_80 | fprs: 20MHz|frt: 240KHz|fsub: 0KHz

<1>

<2>

(c) プロジェクトを読み込んでいる状態 (V850 系 Applilet2)

V850ESJJ3: uPD70F3746 | CPUシステム・クロック: 0.5MHz|メイン・システム・クロック: 4MHz|サブクロック発振回路クロック: 32.768kHz | RTOS: None

<1>

<2>

<3>

備考 <1> デバイス名

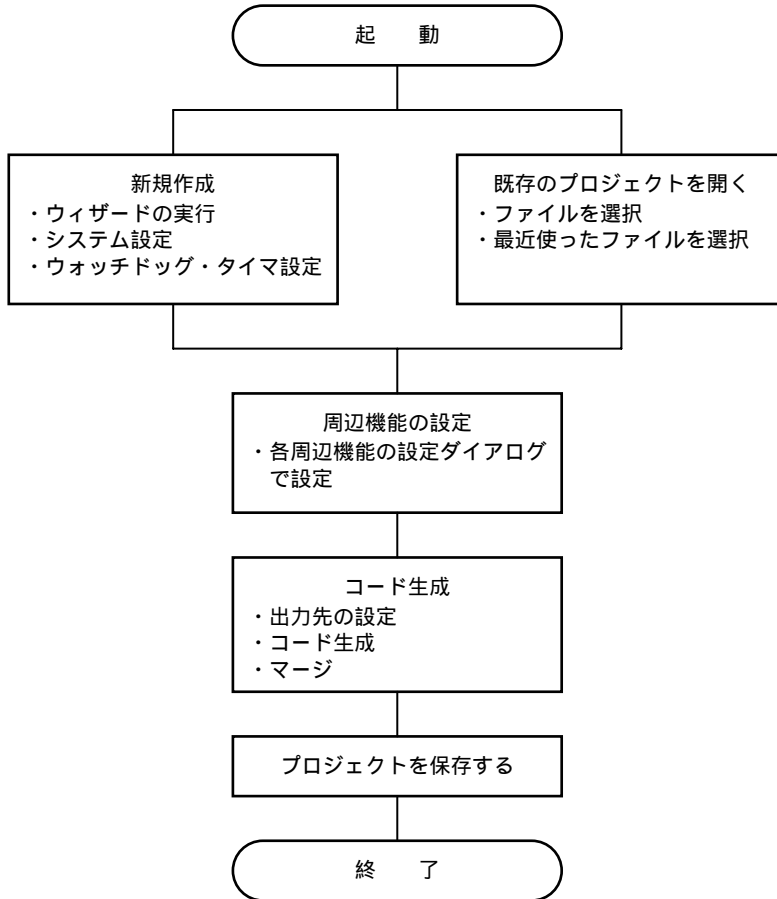
<2> クロック設定情報

<3> RTOS 情報 (Applilet2 V2.41 では、常に “None” を表示します。)

3.2 操作手順

Appliet2 では、次の手順でソース・コードを作成します。

図 3 - 7 操作手順



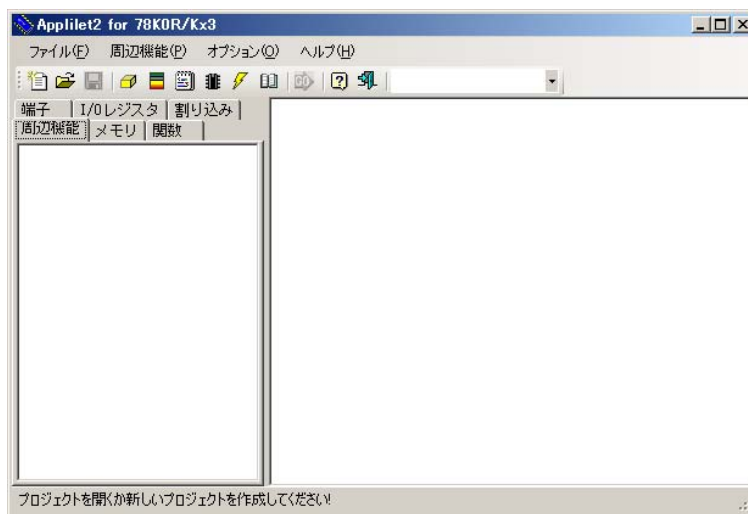
3.3 起 動

Appliet2 を起動するには、2 通りの方法があります。ここでは、NEC Electronics Tools フォルダにインストールした場合を例として説明します。

Windows の [スタート] ボタン [プログラム] [NEC Electronics Tools] [Vx.xx] [Appliet2 for xxxxx Vx.xx] を選択します。選択後、Appliet2 メイン・ウインドウが起動します。

Windows の [スタート] ボタン [プログラム] [NEC Electronics Tools] [最新版] [Appliet2 for xxxxx Vx.xx] を選択します。選択後、Appliet2 メイン・ウインドウが起動します。

図 3 - 8 メイン・ウインドウ (起動直後)




3.4 新規作成

新規にプロジェクト・ファイルを作成するときは、ウィザードを使用します。

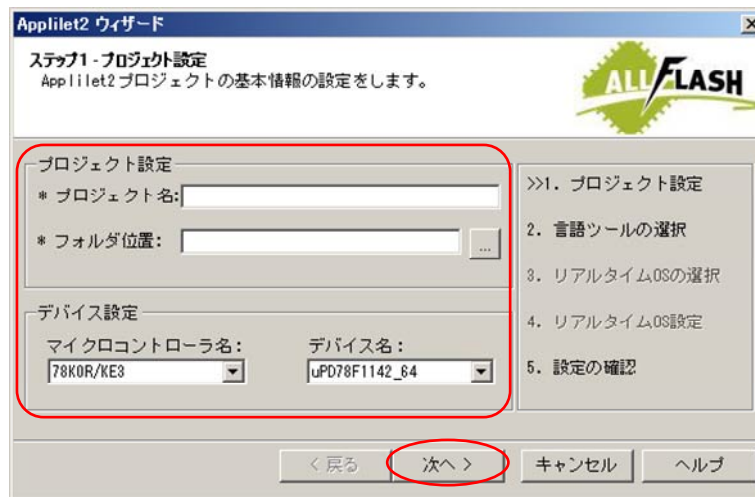
ウィザードでは、プロジェクト・ファイル作成時に、ファイル名、ターゲット・デバイス、言語ツールを選択します。また、これらの言語ツールは、起動後も変更することができます。

ウィザード設定手順を次に説明します。

(1)メイン・ウインドウの [ファイル(F)] メニュー [新規作成(N)...] を選択すると、ウィザードが起動します。または、ツール・バーの  ボタンを押すと、ウィザードが起動します。

(2)ウィザードのステップ1で、プロジェクト名と使用デバイス名を指定します。このプロジェクト名は、コード生成およびビルド後に生成されるオブジェクト・ファイルの名前にも使用されます。指定後、[次へ] ボタンを押します。

図 3-9 ウィザード (ステップ1)



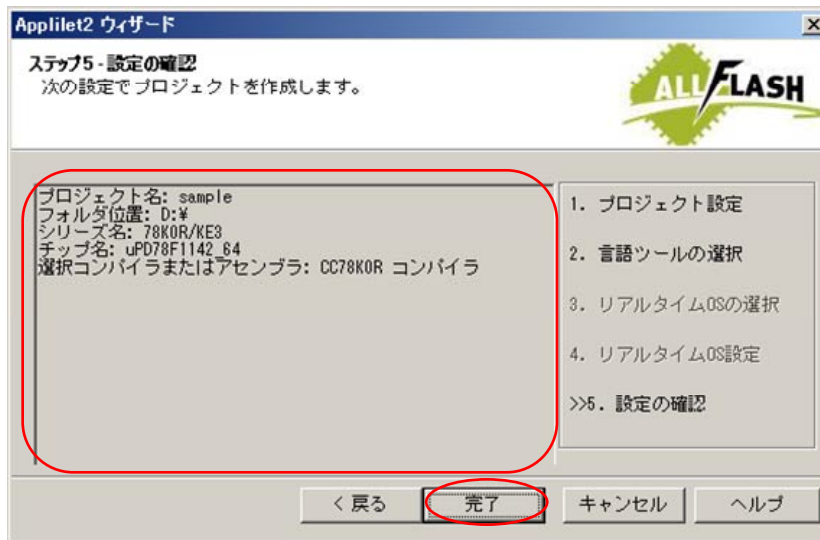
- 備考** 1. [ヘルプ] ボタンを押すとオンライン・ヘルプが表示されます。[キャンセル] ボタンを押すとウィザードを終了し、プロジェクトの作成は中止します。この機能は、各ステップで同じ動作です。
2. 指定したフォルダ位置に同じ名称のプロジェクトが存在する場合、上書きを確認するメッセージが表示されます。[OK] ボタンを押すとウィザードの完了後にプロジェクト・ファイルが上書きされます。

(3)ウィザードのステップ3で、リアルタイム OS を選択します。選択後、[次へ] ボタンを押します (AppIilet2 V2.41 は、リアルタイム OS に対応していません)。

(4)ウィザードのステップ4で、リアルタイム OS で使用するシステム・タイマを選択します。選択後、[次へ] ボタンを押します (AppIilet2 V2.41 は、リアルタイム OS に対応していません)。

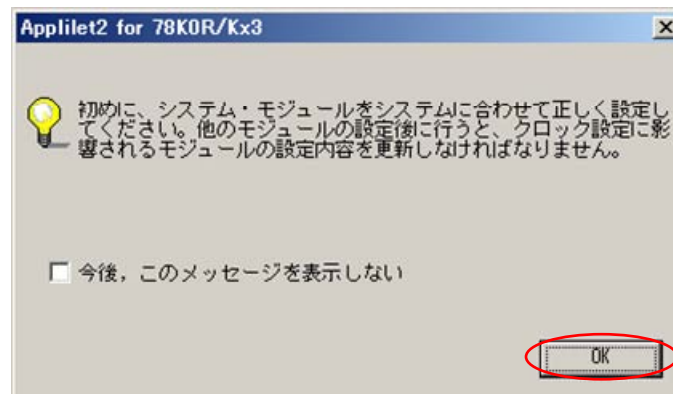
- (5) ウィザードのステップ5で、設定内容を確認します。変更がある場合には、[戻る] ボタンで前のステップへ戻り、変更します。ウィザードを終了するには、[完了] ボタンを押します。

図 3 - 10 ウィザード (ステップ5)



- (6) ウィザード終了後、メッセージ・ダイアログを表示します。このダイアログはいったんチェック ON () にすると次回から表示されません (再インストールをすると再び表示されます)。内容を確認し [OK] ボタンを押します。

図 3 - 11 ウィザード (終了)




- (7) 「システム」ダイアログと「ウォッチドッグ・タイマ」ダイアログ^注を順に表示します。これらの設定を省略して先に進むことはできません。使用するシステムに合わせてこれらのダイアログで設定します。

注 使用するデバイスによっては「ウォッチドッグ・タイマ」ダイアログが表示されない場合があります。

3.5 既存のプロジェクトを開く

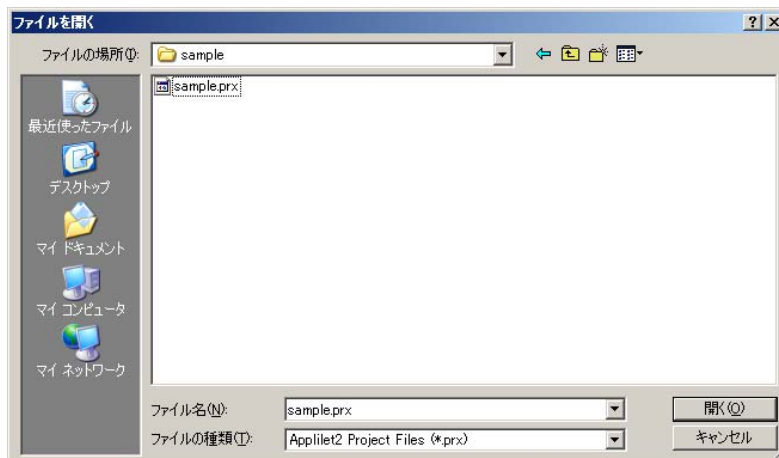
以前に作成したプロジェクトがある場合、次の方法で開きます。

(1) ファイルを指定して開く

メイン・ウィンドウの [ファイル(E)] メニュー [開く(O)...] を選択します。または、ツール・バーの  ボタンを押します。

「ファイルを開く」ダイアログが表示されます。

図 3 - 12 「ファイルを開く」ダイアログ

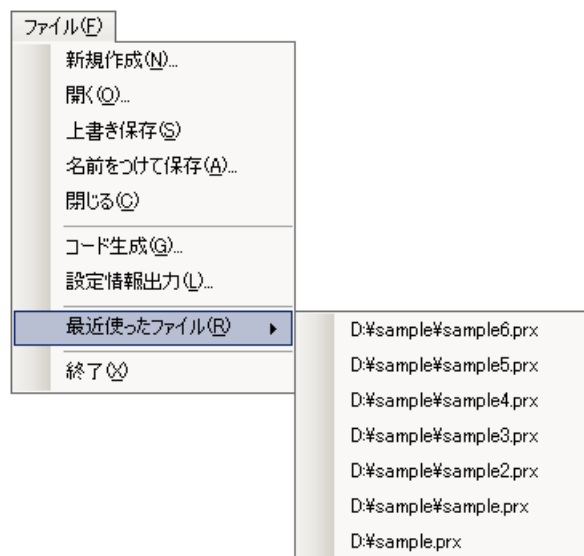


ファイルを選択して [開く] ボタンを押すと選択したファイルを開きます。

(2) 最近使ったファイルを開く

メイン・ウィンドウの [ファイル(E)] メニュー [最近使ったファイル(R)] から、開くファイルを選択します。

図 3 - 13 最近使ったファイルを開く



3.6 周辺機能の設定

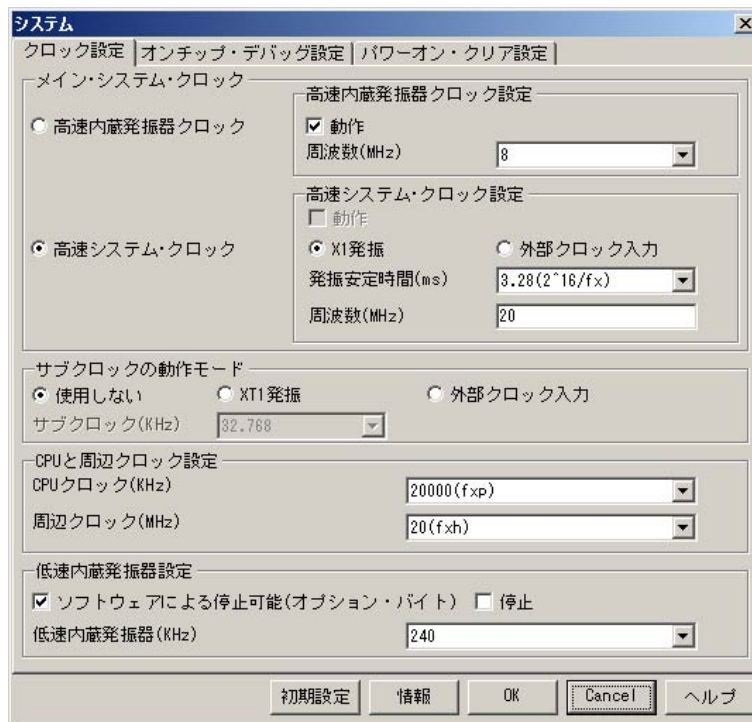
各周辺機能の設定ダイアログは、次のいずれかの方法で表示します。各設定ダイアログの設定については、6.1 周辺機能設定ダイアログを参照してください。

- メイン・ウィンドウの [周辺機能(P)] メニューから周辺機能を選択します。
- ツリー・ビュー・ウィンドウの [周辺機能] タブのアイコンをダブルクリックします。
- 情報表示ウィンドウのアイコンをクリックします。

図 3 - 14 周辺機能の設定



図 3 - 15 設定ダイアログ例 (システム)



注意 「システム」ダイアログの設定は、ほかの周辺機能の設定に影響します。「システム」ダイアログの設定を変更した場合、ほかの周辺機能の設定を再確認する必要があります。

3.7 コード生成

各周辺機能の設定後，コードを生成します。

コード生成は，次のいずれかの方法で表示する「コード生成」ダイアログで行います。


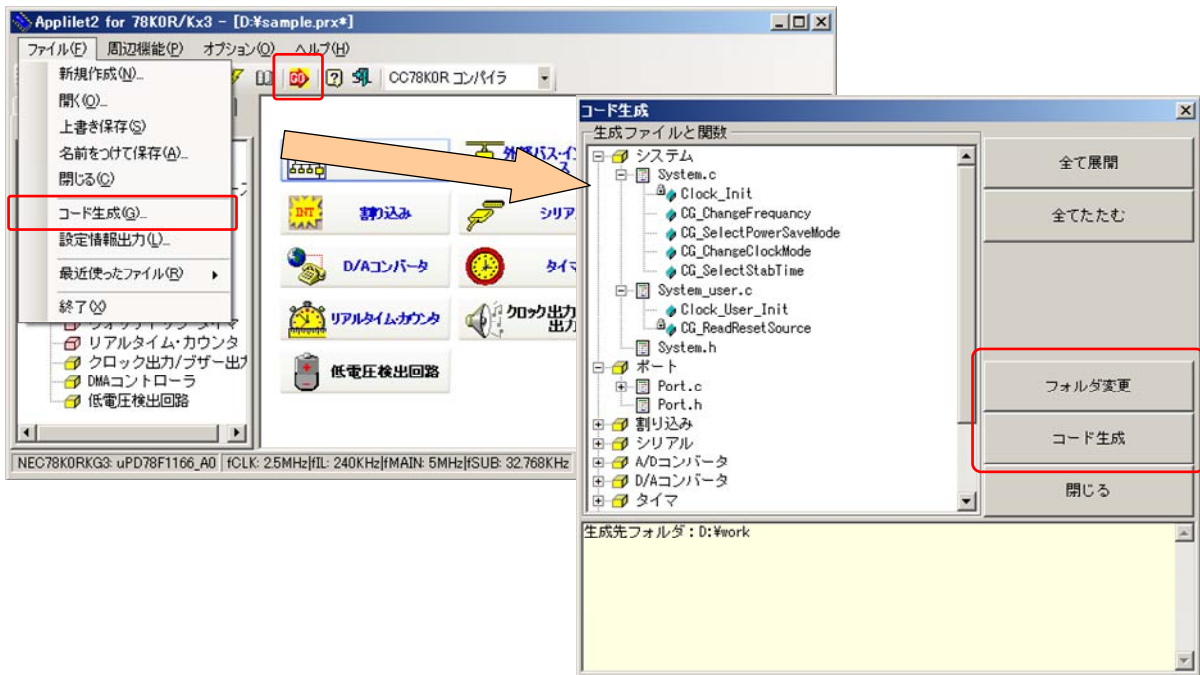
- メイン・ウインドウの [ファイル(E)] メニュー [コード生成(G)...] を選択します。
- ツール・バーの  ボタンを押します。

図 3 - 16 コード生成



「コード生成」ダイアログの詳細については，6.3 「コード生成」ダイアログを参照してください。

- (1) [フォルダ変更] ボタンを押して，コード生成先フォルダを変更できます。
- (2) [コード生成] ボタンを押して，コード生成を実行します。

3.7.1 コード生成先フォルダを変更する

インストール後の初期状態では、コード生成先が Applilet2 のプロジェクト・ファイルが存在するフォルダになっています。[フォルダ変更] ボタンを押すと表示されるダイアログで、コード生成先のフォルダを変更できます。

図 3 - 17 「フォルダの参照」ダイアログ



出力するフォルダの構成は、開発環境により異なります。

表 3 - 1 出力するフォルダの構成

環 境	ファイルの出力先
【NEC 環境】	コード生成先フォルダの直下にすべてのファイルを出力します。
【IAR 環境】	コード生成先フォルダの直下にワーク・スペース・ファイル (xxx.eww) とプロジェクト・ファイル (xxx.ewp) を出力します。コード生成先フォルダの下に " applilet2_src " フォルダを自動で作成し、その他のファイルを出力します。
【GHS 環境】	コード生成先フォルダの直下に上位プロジェクト・ファイル (default_xxx.gpj) を出力します。コード生成先フォルダの下に " applilet2_src " フォルダを自動で作成し、その他のファイルと下位プロジェクト・ファイル (xxx.gpj) を出力します。

備考 各環境用に出力するファイルの構成については、**第7章 出力ファイル**を参照してください。

コード生成先フォルダに同名のファイルが存在しかつ上書きをする場合、バックアップ・ファイルを作成します。バックアップするファイルは、開発環境により異なります。

表 3 - 2 バックアップ・ファイル

環 境	バックアップ・ファイル
【NEC 環境】	コード生成先フォルダの下にバックアップ用の “applilet2_bak” フォルダを作成します。そこへすべてのファイルをバックアップします。上書きをするたびにバックアップ用フォルダ “applilet2_bakn”(n = 1~) を作成します。
【IAR 環境】	“applilet2_src” フォルダの下にバックアップ用の “applilet2_bak” フォルダを作成します。そこへ “applilet2_src” フォルダ以下のファイルをバックアップします。コード生成先フォルダの直下のワーク・スペース・ファイル (xxx.eww) とプロジェクト・ファイル (xxx.ewp) はバックアップしません。上書きをするたびにバックアップ用フォルダ “applilet2_bakn”(n = 1~) を作成します。
【GHS 環境】	“applilet2_src” フォルダの下にバックアップ用の “applilet2_bak” フォルダを作成します。そこへ “applilet2_src” フォルダ以下のファイルをバックアップします。コード生成先フォルダの直下の上位プロジェクト・ファイル (default_xxx.gpj) はバックアップしません。上書きをするたびにバックアップ用フォルダ “applilet2_bakn”(n = 1~) を作成します。

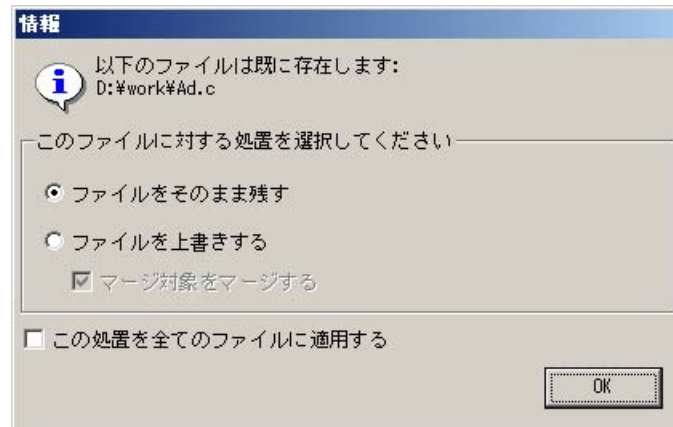
図 3 - 18 コード生成先フォルダのイメージ



3.7.2 上書き確認

ファイル出力時、コード生成先のフォルダに同名のファイルがすでに存在していた場合、次のダイアログを表示します。

図 3 - 19 コード生成時の「情報」ダイアログ (上書き確認)



- [ファイルをそのまま残す]
ファイルを出力せずに既存のファイルを残します。
- [ファイルを上書きする]
ファイルを出力し、既存のファイルに上書きします。
既存のファイルは、バックアップ用フォルダにバックアップとして残します。
- [マージ対象をマージする]
“ファイルを上書きする”を選択すると、この項目が選択可能になります。チェック ON (☑) にすると、既存ファイル内のマージ対象部分を取り出し、新しいファイル内の該当個所に挿入 (マージ) します。
既存のファイルは、バックアップ用フォルダにバックアップとして残します。
マージについては、**3.7.3 マージ機能**を参照してください。
- [この処置を全てのファイルに適用する]
チェック ON (☑) にすると、すべてのファイルに対して現在の選択内容と同じ処置を適用します。以降のファイルに対しては、確認のダイアログを表示しません。

3.7.3 マージ機能

マージ機能は、マージ用コメント内の記述を保護し、ファイルの上書き時に新規に生成するファイルにマージ（コピー）する機能です。

(1) マージ用コメントがある箇所

Applilet2 が出力するファイル内にマージ用コメントがあります。通常は次の箇所に存在します。

- インクルード、グローバル変数定義をしている箇所
- ファイルの最後
- main 関数内、xxx_UserInit 関数内、割り込み関数内

(2) マージ用コメント例

```
/* Start user code. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

注意 1. コメント文は記述されている箇所により異なります。

2. マージ用コメントを編集または移動しないでください。編集または移動した場合、マージが正しく行われません。

(3) マージの実行方法

再コード生成時、「情報」ダイアログ（上書き確認）で“マージ対象をマージする”をチェック ON (☑) にしてコード生成を行うと、マージ用コメント内の記述内容を新しく生成するファイル内の該当箇所にマージ（コピー）します。

(4) マージの実行結果例

マージ用コメント内に記述された内容のみがマージの対象となります。

図 3 - 20 マージの実行結果例

・既存の main.c

```
void main(void)
{
    UCHAR *p_adr = BUFFER;
    /* Start user code. Do not edit comment generated here */
    while(1){
        TMO_Start();
        AD_Start();
    }
    /* End user code. Do not edit comment generated here */
}
```

・再コード生成後の main.c

```
void main(void)
{
    /* Start user code. Do not edit comment generated here */
    while(1){
        TMO_Start();
        AD_Start();
    }
    /* End user code. Do not edit comment generated here */
}
```

マージ用コメント内の記述をマージします。

マージ用コメント外の記述はなくなります。

備考 既存ファイル内に存在する関数（API）が再コード生成時に出力しない設定になっている場合，その関数にマージ用コメントの記述があっても記述内容はなくなります。

図 3 - 21 出力しない関数のマージ例

・既存の AD_user.c

```
      :  
      :  
__interrupt void MD_INTAD(void)  
{  
    /* Start user code. Do not edit comment generated here */  
    AD_Read(DATA_BUF);  
    /* End user code. Do not edit comment generated here */  
}  
      :
```

・A/D コンバータの割り込みの使用をやめて再コード生成した AD_user.c

```
      :  
      :  
      :  
      :
```

割り込み使用をやめることで，MD_INTAD() の出力がなくなります。これにより，該当箇所がなくなるため，MD_INTAD() のマージ用コメント内の記述をマージしません。

3.8 プロジェクトを保存する

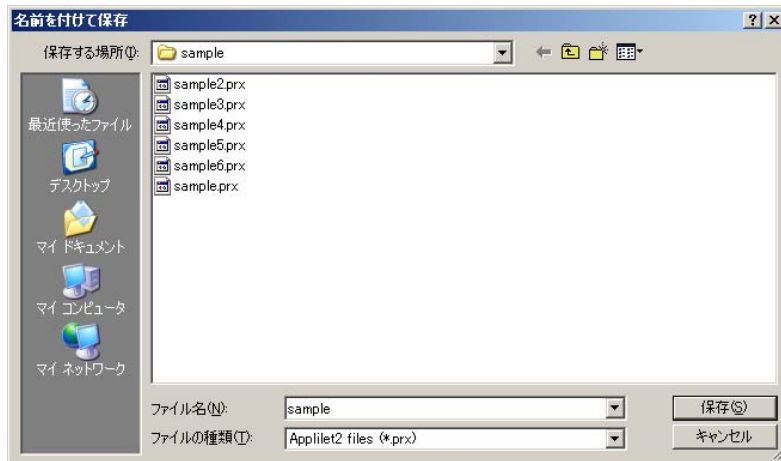
設定した情報を保存するには次の方法があります。

(1) ファイル名を指定して保存する

[ファイル(E)] メニュー [名前をつけて保存(A)...] を選択します。


「名前をつけて保存」ダイアログが表示されます。

図 3 - 22 「名前をつけて保存」ダイアログ



保存先とファイル名を指定し、[保存(S)] ボタンを押すと設定した情報が保存されます。

(2) 上書き保存する

[ファイル(E)] メニュー [上書き保存(S)] を選択します。または、ツール・バーの  ボタンを押します。

編集しているファイル（プロジェクト）が上書き保存されます。

(3) 閉じるときに保存する

設定を変更したあと、保存せずに Appilet2 を終了しようとするとき、保存の確認ダイアログが表示されます。

[はい(Y)] ボタンを押すと編集しているファイル（プロジェクト）が上書き保存されます。

[いいえ(N)] ボタンを押すと設定内容が保存されません。



図 3 - 23 保存の確認ダイアログ



備考 保存するファイル名（拡張子を除く）と Appilet2 のプロジェクト名は同じです。

3.9 終了

Applilet2 を終了するには、次の方法があります。

- メイン・ウィンドウの [ファイル(F)] メニュー [終了(X)] を選択します。
- ツール・バーの  ボタンを押します。
- メイン・ウィンドウの  ボタンを押します。
- タイトル・バーのアイコンをクリックしたときに表示されるメニューで、[閉じる(C)] を選択します。

3.10 コーディング

コード生成後、出力したワーク・スペース/プロジェクト・ファイルを統合開発環境プラットフォームで読み込みます。

ユーザ・ソース・ファイルを追加します。または、必要に応じて Applilet2 が出力したファイル中のマージ用コメント内にコードを追加してプログラムを完成させます。

Applilet2 が出力したワーク・スペース/プロジェクト・ファイルを統合開発環境プラットフォームで読み込んでいる状態で、Applilet2 で再設定、再コード出力を行う場合、次の手順で行います。

- NEC 環境
Applilet2 で再コード出力後、ワーク・スペース・ファイル (.prw) を再読み込みします。
- IAR 環境
IAR Embedded Workbench で、ワーク・スペース・ファイル (.eww) を保存します。
Applilet2 で再コード出力後、ワーク・スペース・ファイル (.eww) を再読み込みします。
- GHS 環境
MULTI で、プロジェクト・ファイル (.gpj) を保存します。
Applilet2 で再コード出力後、プロジェクト・ファイル (.gpj) を再読み込みします。

第4章 メニュー・リファレンス

各メニューとツール・バーの機能について説明します。

4.1 [ファイル(F)] メニュー

図 4 - 1に [ファイル(F)] メニューの表示例を示します。

表 4 - 1に [ファイル(F)] メニューの機能を示します。

図 4 - 1 [ファイル(F)] メニュー

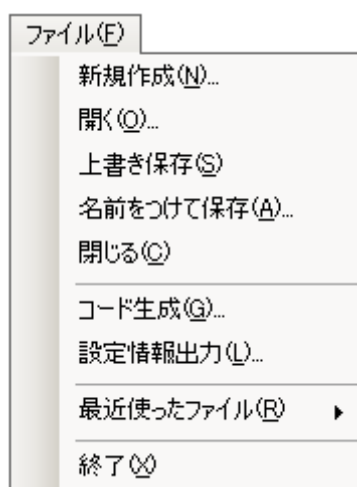


表 4 - 1 [ファイル(F)] メニュー

項 目	内 容
[新規作成(N)...]	新規のプロジェクトを作成や、ウィザードを表示します。
[開く(O)...]	既存のプロジェクトを読み込みます。
[上書き保存(S)]	現在読み込んでいるプロジェクトに、現在の設定内容を上書き保存します。
[名前をつけて保存(A)...]	現在の設定内容を別名のプロジェクトに保存します。
[閉じる(C)]	現在読み込んでいるプロジェクトを終了します。
[コード生成(G)...]	コード生成します。「コード生成」ダイアログを表示します。
[設定情報出力(L)...]	設定情報をファイルに出力します。「設定情報出力」ダイアログを表示します。
[最近使ったファイル(R)]	最近読み込んだプロジェクトを表示します。プロジェクトを選択するとそのプロジェクトを読み込みます。
[終了(X)]	Applilet2を終了します。

4.2 [周辺機能(P)] メニュー

[周辺機能(P)] メニューには、ターゲット・デバイスが持つ周辺機能（Applilet2 でサポートしている周辺機能のみ）を表示します。

図 4 - 2に [周辺機能(P)] メニューの表示例を示します。

表 4 - 2に [周辺機能(P)] メニューの機能を示します。

図 4 - 2 [周辺機能(P)] メニュー

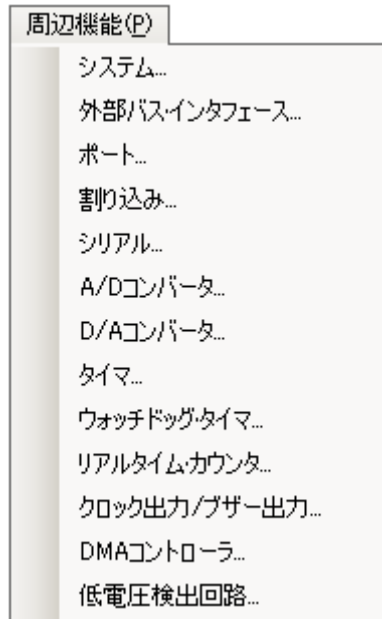


表 4 - 2 [周辺機能(P)] メニュー

項 目	内 容
周辺機能名	選択した周辺機能の設定ダイアログを表示します。

4.3 [オプション(O)] メニュー

図 4 - 3に [オプション(O)] メニューの表示例を示します。

表 4 - 3に [オプション(O)] メニューの機能を示します。

図 4 - 3 [オプション(O)] メニュー

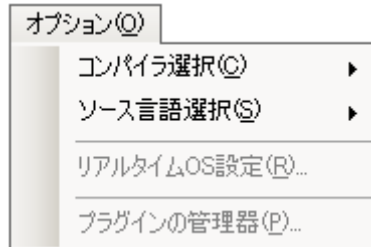


表 4 - 3 [オプション(O)] メニュー

項 目	内 容
[コンパイラ選択(C)]	出力コードの形式を選択します。表示するコンパイラ名は、製品により異なります。 【78K0系 Applilet2, 78K0R系 Applilet2】 [NECコンパイラ] : NECコンパイラ形式のコードを出力します。 [IARコンパイラ] : IARコンパイラ形式のコードを出力します。 【V850系 Applilet2】 [NECコンパイラ] : NECコンパイラ形式のコードを出力します。 [IARコンパイラ] : IARコンパイラ形式のコードを出力します。 [GHSコンパイラ] : GHSコンパイラ形式のコードを出力します。
[ソース言語選択(S)]	出力コードの言語を選択します。V850系製品では、言語の選択はできません。 【78K0S, 78K0 [※] , 78K0R】 [C言語] : Cソースのコードを出力する。 [ASM言語] : アセンブラ・ソースのコードを出力する。 注 78K0/LX3用では、[ASM言語]は選択不可です。
[リアルタイムOS設定(R)...]	選択不可。
[プラグインの管理者(P)...]	選択不可。

4.4 [ヘルプ(H)] メニュー

図 4 - 4に [ヘルプ(H)] メニューの表示例を示します。

表 4 - 4に [ヘルプ(H)] メニューの機能を示します。

図 4 - 4 [ヘルプ(H)] メニュー

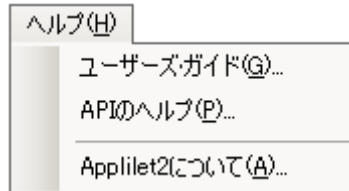


表 4 - 4 [ヘルプ(H)] メニュー

項 目	内 容
[ユーザーズ・ガイド(G)...]	ヘルプ (ユーザーズ・ガイド) のトップ・ページを表示します。
[API のヘルプ(P)...]	ヘルプ (API) のトップ・ページを表示します。
[Applet2 について(A)...]	「Applet2 について」ダイアログを表示します。

4.5 ツール・バー

ツール・バーは、メニュー・バーの下に常時表示しているツール・バー（ツール・バー1）と、ツリー・ビュー・ウィンドウにおいて、[周辺機能] タブ以外が選択されているときに、メイン・ウィンドウの右横に表示するツール・バー（ツール・バー2）があります。

4.5.1 ツール・バー1

ツール・バー1には、次のボタンおよびリスト・ボックスがあります。

図 4-5 ツール・バー1



表 4-5 ツール・バー1

項目	内容
	新規プロジェクトを作成します。ウィザードを表示します。
	既存のプロジェクトを読み込みます。
	現在読み込んでいるプロジェクトに現在の設定内容を上書き保存します。
	ツリー・ビュー・ウィンドウに [周辺機能] タブを表示します。
	ツリー・ビュー・ウィンドウに [メモリ] タブを表示します。
	ツリー・ビュー・ウィンドウに [関数] タブを表示します。
	ツリー・ビュー・ウィンドウに [端子] タブを表示します。
	ツリー・ビュー・ウィンドウに [割り込み] タブを表示します。
	ツリー・ビュー・ウィンドウに [I/O レジスタ] タブを表示します。
	「コード生成」ダイアログを表示します。
	ヘルプを表示します。
	Applilet2 を終了します。
CC78K0R コンパイラ	出力コードのコンパイラと言語を選択します。 選択可能なコンパイラと言語は、製品により異なります。

4.5.2 ツール・バー2



ターゲット・デバイスが持つ、周辺機能（Applilet2 でサポートしている周辺機能のみ）を表すアイコンを表示します。

図 4 - 6 ツール・バー2



アイコンをクリックすると設定ダイアログを表示します。

設定の状態に合わせてアイコンの色が変わります。

-  (黒色): 機能を設定済 (設定ダイアログにおいて、OK ボタンで 1 回以上閉じたことがある)
-  (黄色): 機能が未設定 (設定ダイアログを未表示、または OK ボタンで 1 回も閉じたことがない)

アイコン上でマウスの右ボタンをクリックするとコンテキスト・メニューを表示します。

表 4 - 6 コンテキスト・メニュー

項目	内容
[初期設定(D)]	選択している周辺機能の設定を Applilet2 の初期設定の状態に戻します。 設定ダイアログにタブがある場合、すべてのタブおよびサブ・ダイアログの内容を初期設定状態に戻します。

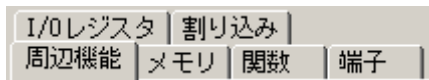
第5章 ウィンドウ・リファレンス

Applilet2 は、対象となるマイクロコントローラ製品ごとにウィンドウが異なります。

本マニュアルでは、対象となるマイクロコントローラ製品に共通する表示および使用方法について説明します。製品ごとのウィンドウの説明は省略しています。

ウィンドウの表示内容は、選択しているツリー・ビュー・ウィンドウのタブに合わせて切り替わります。

図 5 - 1 ツリー・ビュー・ウィンドウのタブ



5.1 [周辺機能] タブ

ターゲット・デバイスが持つ、周辺機能（Applilet2 でサポートしている周辺機能のみ）を表示します。

図 5 - 2 [周辺機能] タブ





(1) ツリー・ビュー・ウィンドウ ([周辺機能] タブ)

ターゲット・デバイスが持つ、周辺機能（Applilet2 でサポートしている周辺機能のみ）をツリー形式で表示します。

ツリー・ビュー・ウィンドウ上の各周辺機能をダブルクリックすると設定ダイアログを表示します。

設定の状態に合わせてアイコンの色が変わります。

 (桃色)： 機能を設定済（設定ダイアログにおいて、OK ボタンで 1 回以上閉じたことがある）

 (黄色)： 機能が未設定（設定ダイアログを未表示、または OK ボタンで 1 回も閉じたことがない）

(2) 情報表示ウィンドウ ([周辺機能] タブ)

ターゲット・デバイスが持つ、周辺機能 (Applilet2 でサポートしている周辺機能のみ) を表示します。

アイコンをクリックすると設定ダイアログを表示します。

設定の状態に合わせてアイコンの文字列の色が変わります。



シリアル

(黒色): 機能を設定済

(設定ダイアログにおいて、OK ボタンで 1 回以上閉じたことがある)



シリアル

(青色): 機能が未設定

(設定ダイアログを未表示、または OK ボタンで 1 回も閉じたことがない)

アイコン上でマウスの右ボタンをクリックするとコンテキスト・メニューを表示します。

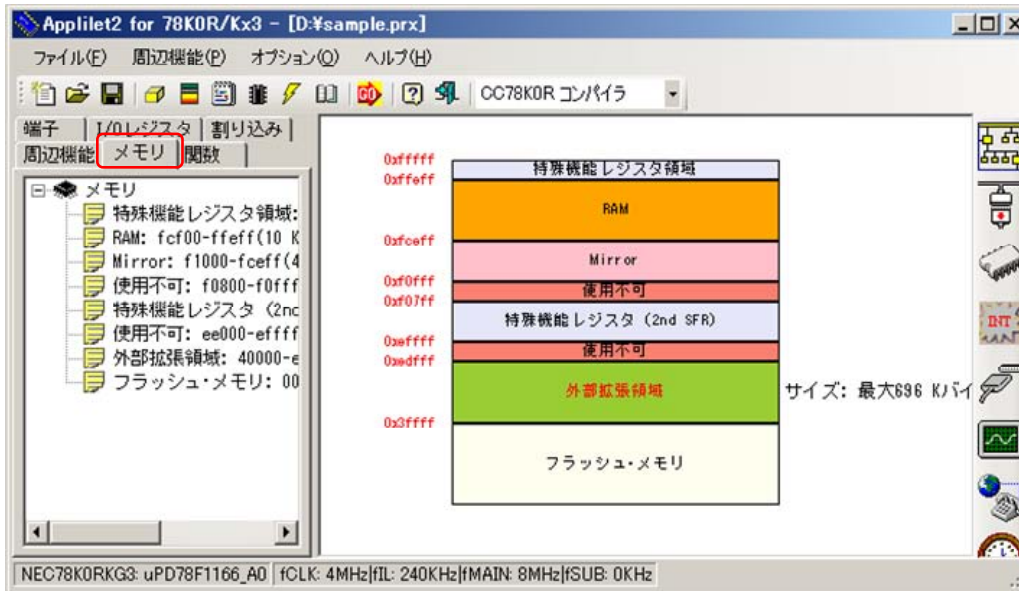
表 5 - 1 情報表示ウィンドウのコンテキスト・メニュー ([周辺機能] タブ)

項 目	内 容
[初期設定(D)]	選択している周辺機能の設定を Applilet2 の初期設定の状態に戻します。 設定ダイアログにタブがある場合、すべてのタブおよびサブ・ダイアログの内容を初期設定状態に戻します。

5.2 [メモリ] タブ

ターゲット・デバイスのメモリ空間を表示します。

図 5-3 [メモリ]タブ



(1) ツリー・ビュー・ウィンドウ ([メモリ] タブ)

ターゲット・デバイスのメモリ空間を、領域別にツリー形式で表示します。

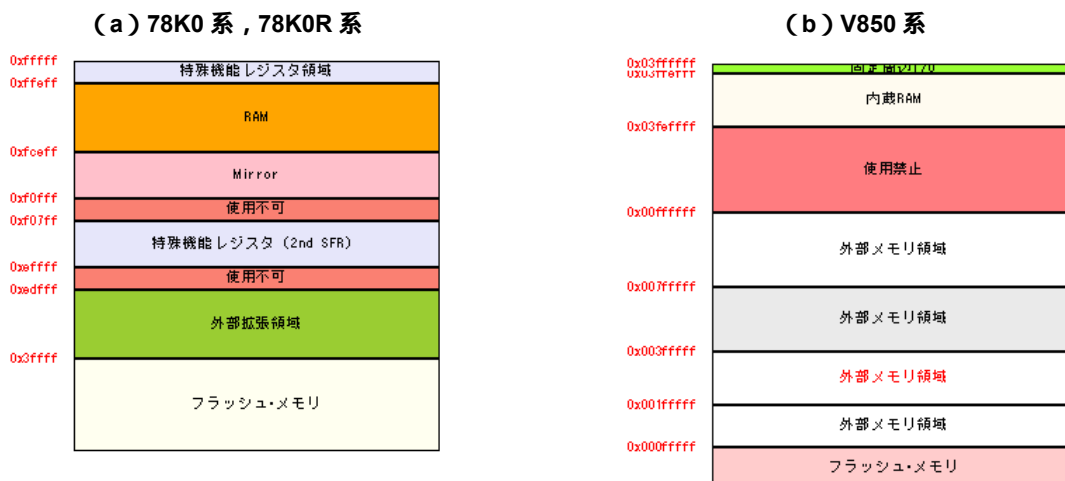
(2) 情報表示ウィンドウ ([メモリ] タブ)

ターゲット・デバイスのメモリ空間 (メモリ・マップのイメージ) を表示します。

メモリ・マップ上にマウス・カーソルを移動すると、該当する領域のサイズをメモリ・マップの右側に表示します。

V850 系 AppIlet2 では、外部バス・インタフェースの設定ダイアログの設定に合わせて、外部メモリを使用する領域の色が変化します。

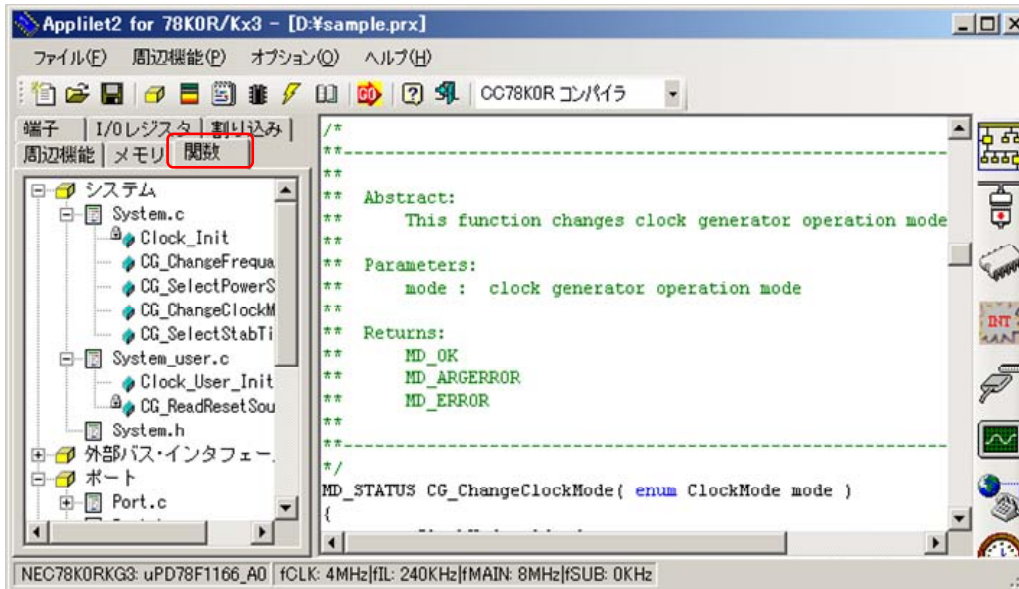
図 5-4 メモリ空間の表示例



5.3 [関数] タブ

Applilet2 が出力するファイル名や API 名およびファイルの内容を表示します。

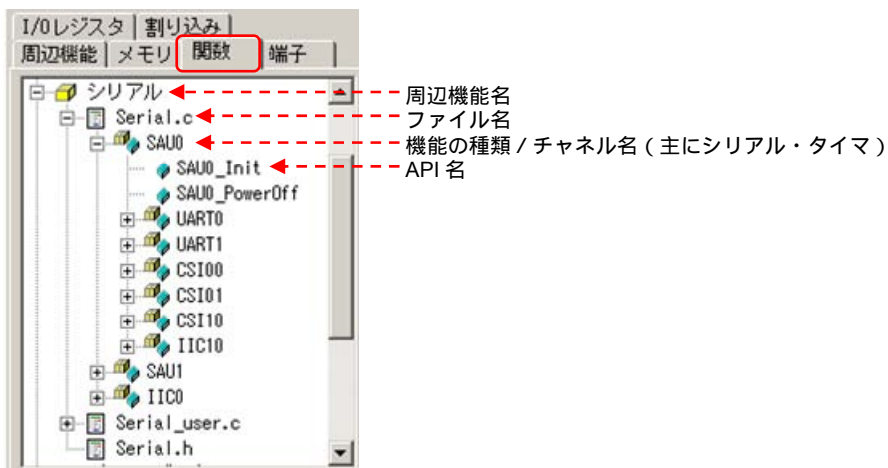
図 5-5 [関数] タブ






(1) ツリー・ビュー・ウィンドウ ([関数] タブ)

Applilet2 が出力するファイル名や API 名を周辺機能別にツリー形式で表示します。
 ファイル名や API 名を選択すると、情報表示ウィンドウにファイルの内容が表示されます。

図 5-6 ツリー・ビュー・ウィンドウ ([関数] タブ)



API 名のアイコンは、次の種類があります。

-  : コード生成時に API を出力しません。
-  : コード生成時に API を出力します(現在の周辺機能の設定では出力する必要はありませんが、ユーザが出力する設定に変更しています)。
-  : コード生成時に API を出力します(現在の周辺機能の設定では出力する必要があります。ユーザが出力しない設定に変更できません)。

⚡ (黄色) : 割り込み関数です。コード生成時に出力しません。

🍑 (桃色) : 割り込み関数です。コード生成時に出力します。

ファイル名および API 名上でマウスの右ボタンをクリックするとコンテキスト・メニューを表示します。

表 5 - 2 ファイル名のコンテキスト・メニュー

項目	内容
[リネーム(R)]	ファイル名を変更します。 選択すると、ファイル名の表示が変更可能なテキスト・ボックスに変化します。
[初期設定(D)]	ファイル名を Applilet2 の初期設定の名前に戻します。

表 5 - 3 API 名のコンテキスト・メニュー

項目	内容
[出力する]	コード生成時に API を出力します。 アイコンが 🍑 または ⚡ の API のみを選択可能です。
[出力しない]	コード生成時に API を出力しません。 アイコンが 🍑 または ⚡ の API のみを選択可能です。
[リネーム(R)]	API 名を変更します。 選択すると、API 名の表示が変更可能なテキスト・ボックスに変化します。
[初期設定(D)]	API 名を Applilet2 の初期設定の名前に戻します。

(2) 情報表示ウィンドウ ([関数] タブ)

Applilet2 が出力するファイルの内容をプレビューします。

ツリー・ビュー・ウィンドウでファイル名をクリックすると、ファイルの先頭部分を表示します。API 名をクリックするとその API の内容を表示します。

注意 プレビュー内容は、現在の周辺機能の設定に従った内容を表示します。このため、プレビューと実際に出力されるコードが同一にならない場合があります。

図 5 - 7 API 表示例

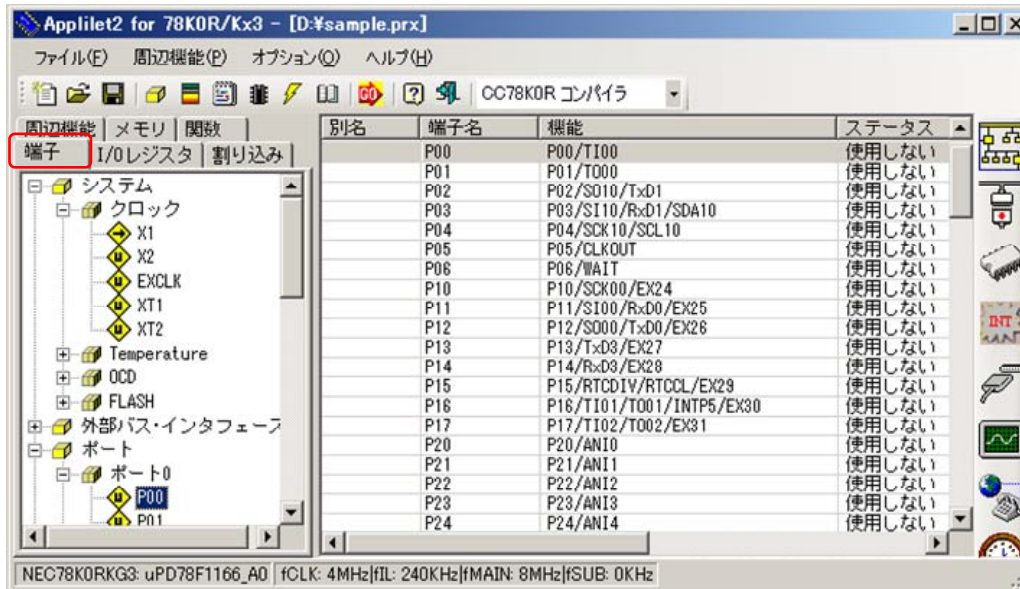
```
void INTP_Init( void )
{
    EGP0 = INTP_EGP_RESET_VALUE;
    EGN0 = INTP_EGN_RESET_VALUE;
    EGP1 = INTP_EGP_RESET_VALUE;
    EGN1 = INTP_EGN_RESET_VALUE;
    INTP_User_Init(); ←
}
```

このAPI (INTP_Init()) を出力した場合、この位置で関数をコールします。
コード生成時、実際にコードを出力する / しないは、ツリー・ビュー・ウィンドウのこのAPI (INTP_Init()) の設定に従います。

5.4 [端子] タブ

ターゲット・デバイスが持つ端子の機能と設定状態を表示します。

図 5-8 [端子] タブ



(1) ツリー・ビュー・ウィンドウ ([端子] タブ)

ターゲット・デバイスが持つ端子を、周辺機能別にツリー形式で表示します。

ツリー・ビュー・ウィンドウ上の端子名をクリックすると、情報表示ウィンドウでクリックした端子名の箇所を表示します。

アイコンは、次の種類があります。

- ◀ : 現在の設定では未使用の端子です。
- ▶ : 現在の設定で使用する端子です。入出力は出力です。
- ◀▶ : 現在の設定で使用する端子です。入出力は入力です。
- ◀▶ : 現在の設定で使用する端子です。入出力は入出力です。

(2) 情報表示ウィンドウ ([端子] タブ)

ターゲット・デバイスが持つ端子を表形式で表示します。

表の各項目は次の内容を示します。

表 5 - 4 情報表示ウィンドウの表示項目 ([端子] タブ)

項 目	内 容
別名	端子名の別名を表示します。別名がない場合、空欄になります。セルをダブルクリックすると各端子の別名を編集できます。
端子名	ポート端子名 (または兼用機能がない端子名) を表示します。
機能	兼用機能を含めた端子名を表示します。
ステータス	設定している機能名を表示します。 機能が設定されていない場合、“ 使用しない ” を表示します。
I/O	現在の設定で使用する端子の入出力を表示します。使用しない端子は空欄になります。 I : 入力端子 O : 出力端子 I/O : 入出力端子 - : Applilet2 上で入出力を区別しない端子 (電源系の端子など)
備考	設定している端子機能の簡単な説明を表示します。使用しない端子は空欄になります。

項目名をクリックすると表示内容を並び替えます。初期状態の並び順は不定です。いったん並び替えると初期状態の並び順には戻せません。

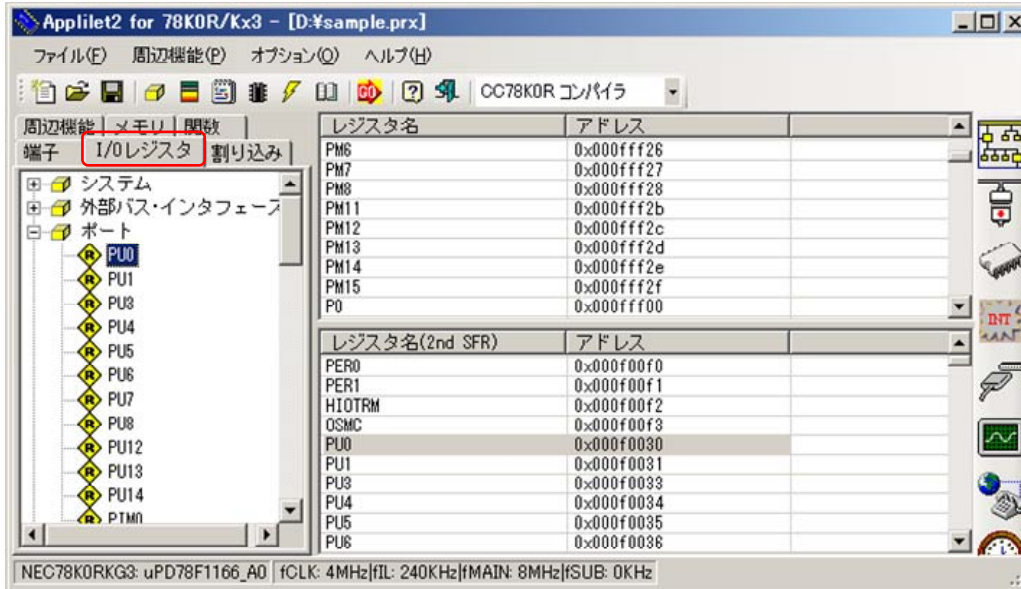
表 5 - 5 項目名と並び順 ([端子] タブ)

項 目	並び順
別名	空欄, 数字順, アルファベット順に並び替えます。
端子名	アルファベット順に並び替えます。
機能	アルファベット順に並び替えます。
ステータス	アルファベット, 未使用の順に並び替えます。
I/O	空欄, -, I, I/O, O の順に並び替えます。
備考	空欄, アルファベット, 2 バイト文字の順に並び替えます。

5.5 [I/O レジスタ] タブ

ターゲット・デバイスが持つ特殊機能レジスタ（78K）や周辺 I/O レジスタ（V850）を表示します。

図 5-9 [I/O レジスタ] タブ




(1) ツリー・ビュー・ウィンドウ ([I/O レジスタ] タブ)

ターゲット・デバイスが持つ特殊機能レジスタ（78K）や周辺 I/O レジスタ（V850）を、周辺機能別にツリー形式で表示します。

ツリー・ビュー・ウィンドウ上のレジスタ名をクリックすると、情報表示ウィンドウでクリックしたレジスタ名の箇所を表示します。

アイコンは、次の種類があります。

 : レジスタを示します。

(2) 情報表示ウィンドウ ([I/O レジスタ] タブ)

ターゲット・デバイスが持つ特殊機能レジスタ（78K）/ 周辺 I/O レジスタ（V850）を表形式で表示します。表の各項目は次の内容を示します。

表 5-6 情報表示ウィンドウの表示項目 ([I/O レジスタ] タブ)

項目	内容
レジスタ名	特殊機能レジスタ（78K）/ 周辺 I/O レジスタ（V850）を表示します。 78K0R の場合、特殊機能レジスタ（SFR）と拡張特殊機能レジスタ（2nd SFR）を別表で表示します。
アドレス	特殊機能レジスタ（78K）/ 周辺 I/O レジスタ（V850）が割り付けられているアドレスを表示します。

項目名をクリックすると表示内容を並び替えます。初期状態の並び順は、ツリー・ビュー・ウィンドウで表示している周辺機能順です。

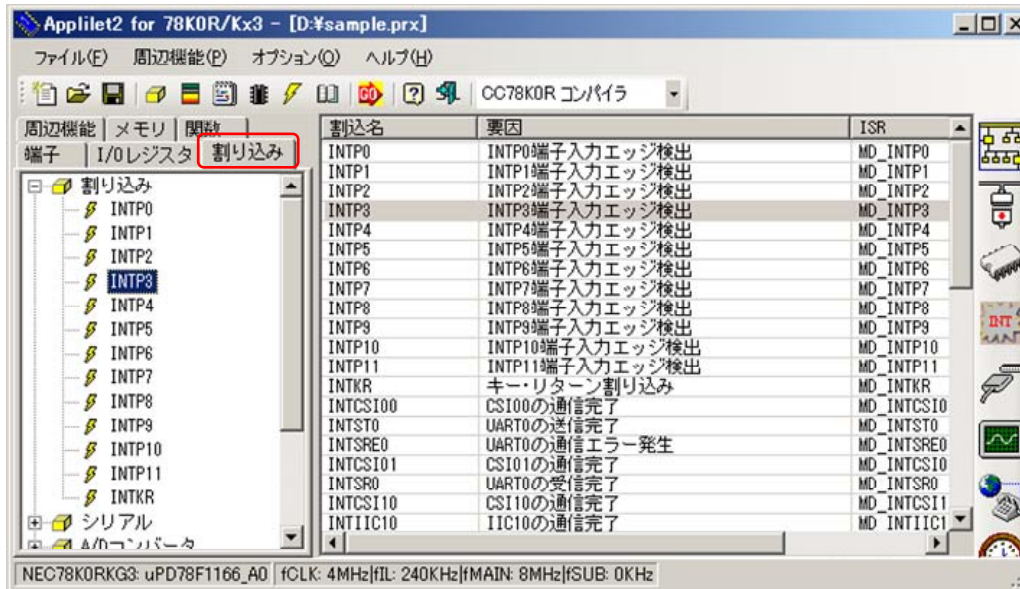
表 5 - 7 項目名と並び順 ([I/O レジスタ] タブ)

項 目	並び順
レジスタ名	アルファベット順に並び替えます。
アドレス	アドレスの小さい順に並び替えます。

5.6 [割り込み] タブ

ターゲット・デバイスが持つ割り込み要因を表示します。

図 5 - 10 [割り込み] タブ



(1) ツリー・ビュー・ウィンドウ ([割り込み] タブ)

ターゲット・デバイスが持つ割り込み要因を、周辺機能別にツリー形式で表示します。

ツリー・ビュー・ウィンドウ上のレジスタ名をクリックすると、情報表示ウィンドウでクリックしたレジスタ名の箇所を表示します。

設定に合わせてアイコンの色が変わります。

⚡ (黄色) : 現在の設定は、未使用の割り込み要因です。

🍑 (桃色) : 現在の設定は、使用する割り込み要因です。

(2) 情報表示ウィンドウ ([割り込み] タブ)

ターゲット・デバイスが持つ割り込み要因を表形式で表示します。

表の各項目は次の内容を示します。

表 5 - 8 情報表示ウィンドウの表示項目 ([割り込み] タブ)

項目	内容
割り込み名	割り込み要因名を表示します。
要因	割り込み要因のトリガを表示します。
ISR	Applilet2 で出力する割り込み関数名を表示します。
ステータス	現在の設定で、使用するかないかを表示します。

項目名をクリックすると表示内容を並び替えます。初期状態の並び順は、ツリー・ビュー・ウインドウで表示している周辺機能順です。いったん並び替えると初期状態の並び順には戻せません。

表 5 - 9 項目名と並び順 ([割り込み] タブ)

項 目	並び順
割り込み名	アルファベット順に並び替えます。
要因	アルファベット順, 2 バイト文字の順に並び替えます。
ISR	アルファベット順に並び替えます。
ステータス	「使用されます」, 「使用しない」の順に並び替えます。

第6章 ダイアログ・リファレンス

Applilet2 は、対象となるマイクロコントローラ製品ごとにダイアログが異なります。

本マニュアルでは、対象となるマイクロコントローラ製品に共通する表示方法および使用方法について説明します。製品ごとのダイアログの説明は省略しています。

6.1 周辺機能設定ダイアログ

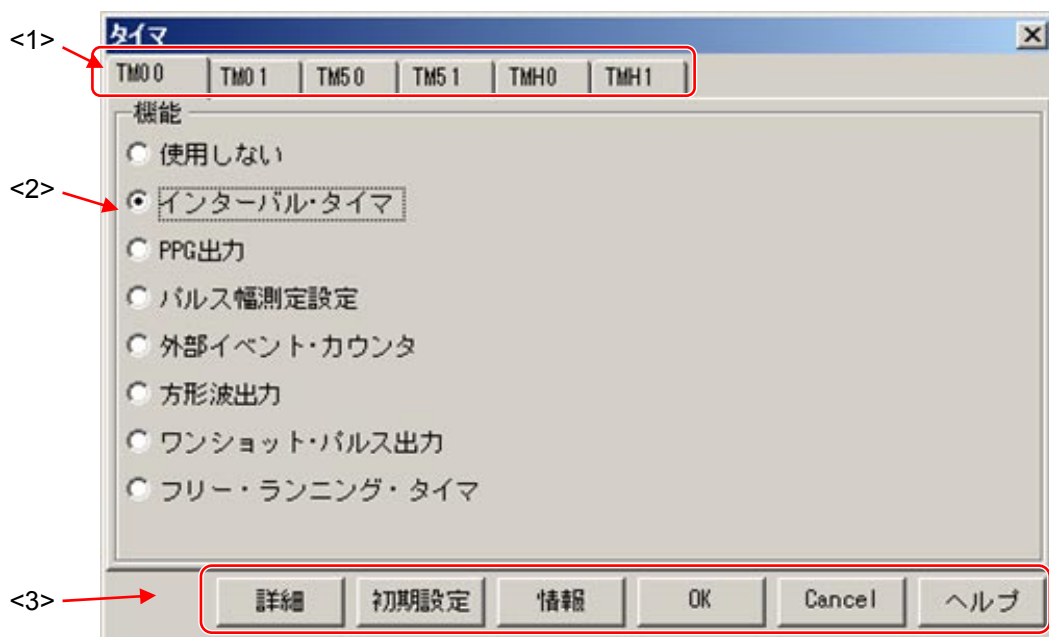
周辺機能ごとの設定ダイアログがあります。

設定ダイアログでは、周辺機能の初期設定を行います。

6.1.1 設定ダイアログの構成

設定ダイアログの構成は周辺機能ごとに異なります。

図 6-1 設定ダイアログの構成



備考 <1> タブ : 設定するチャンネル / 機能を選択します。

周辺機能によっては、設定ダイアログにタブがない場合があります。

<2> 設定項目 : ラジオ・ボタン、テキスト・ボックス、コンボ・ボックスなどで、機能 / モードを設定します。

<3> ボタン : 設定の確定、中止などの操作を行います。

[OK] ボタンを押すと、現在の設定内容を確定し設定ダイアログを閉じます。

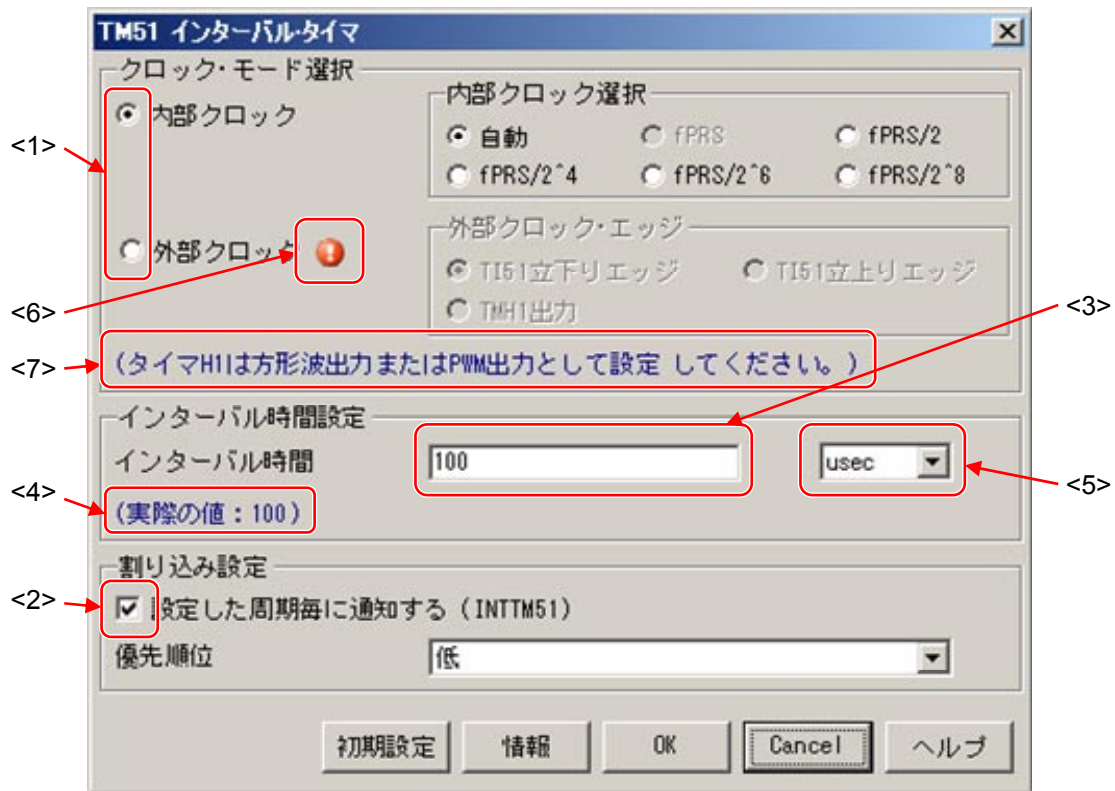
[Cancel] ボタンを押すと、現在の設定内容を確定せずに設定ダイアログを閉じます。

[ヘルプ] ボタンを押すと、オンライン・ヘルプを表示します。

6.1.2 設定方法

設定ダイアログ内の各項目の設定方法について説明します。

図 6-2 設定ダイアログの設定項目



- 備考**
- <1> ラジオ・ボタン (/)
 - <2> チェック・ボックス (/)
 - <3> テキスト・ボックス
 - <4> 計算値メッセージ
 - <5> コンボ・ボックス
 - <6> 警告マーク (端子競合)
 - <7> インフォメーション

(1) ラジオ・ボタン (/)

同じカテゴリ内の選択肢の中から排他的に 1 つを選択します。

(2) チェック・ボックス (/)

機能の有効 / 無効を設定します。

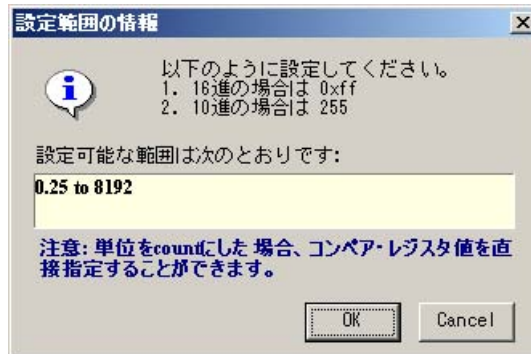
チェック ON () にすると有効になります。

(3) テキスト・ボックス

数値を入力します。10進数、16進数、または小数を入力できます。

テキスト・ボックス内をダブルクリックすると、入力可能な値の範囲を示すダイアログを表示します。

図 6-3 テキスト・ボックスの設定範囲の情報





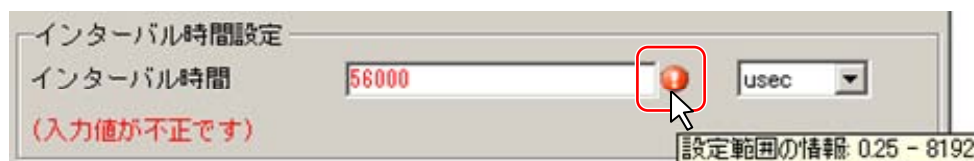
入力値が範囲を超えると文字列を赤で表示します。また、 (警告マーク) を表示します。 上にマウス・カーソルを合わせると、入力可能な範囲をマウス・カーソル付近に表示します。

図 6-4 警告マーク (範囲外) の設定範囲の情報



(4) 計算値メッセージ

現在の周辺機能のクロック設定とレジスタ設定値を元に論理的に計算した値 (実際の値 / 誤差) を表示します。

計算値メッセージは、一部のテキスト・ボックスまたはコンボ・ボックスにのみ存在します。

主に次のメッセージがあります。

- (実際の値: x x x)
- (誤差: x.xx%)

テキスト・ボックスまたはコンボ・ボックスに範囲外の値を入力すると、エラー・メッセージに変化します。

- (実際の値: x x x) (入力値が不正です)
- (誤差: x.xx%) (ポーレートが許容範囲を超えています。)

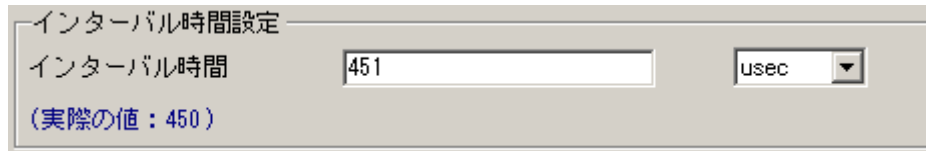
または

(入力値が不正です!)

例 インターバル・タイマにおいて、インターバル時間に $451 \mu\text{s}$ を設定した場合、次の条件下では論理上実現できないため、実際は $450 \mu\text{s}$ となることを示します。

- ・周辺機能のクロック ($f_{\text{PRS}} = 8 \text{ MHz}$)
- ・タイマの内部クロック = $f_{\text{PRS}} / 2^4 = 0.5 \text{ MHz}$ (タイマの 1 カウントは $2 \mu\text{s}$)

図 6-5 計算値メッセージ例



(5) コンボ・ボックス

▼ をクリックすると表示されるリストから設定値を選択します。

一部のコンボ・ボックスでは、リストにない値を入力できる場合があります。たとえば、シリアル・インタフェースの設定ダイアログのボーレート設定では、値を入力できます。

- ・値の選択のみを行えるコンボ・ボックス

項目設定時にボックス全体をハイライト表示します。この場合、計算値メッセージを表示しません。

図 6-6 値の選択のみを行えるコンボ・ボックス



- ・値を入力可能なコンボ・ボックス

項目設定時に文字列の部分のみをハイライト表示します。この場合、計算値メッセージを表示します。

図 6-7 値を入力可能なコンボ・ボックス



値を入力可能なコンボ・ボックス内をダブルクリックすると、入力可能な値の範囲を示すダイアログを表示します。

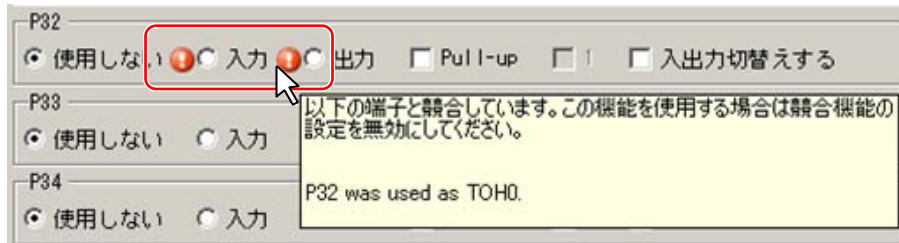
図 6-8 入力可能な値の範囲を示すダイアログ例



(6) 警告マーク (端子競合)

すでに兼用機能または兼用機能端子が使用されていて、設定できないことを示します。警告マーク (端子競合) 上にマウス・カーソルを合わせると、競合している設定をマウス・カーソル付近に表示します。

図 6 - 9 警告マーク (端子競合)

**(7) インフォメーション**

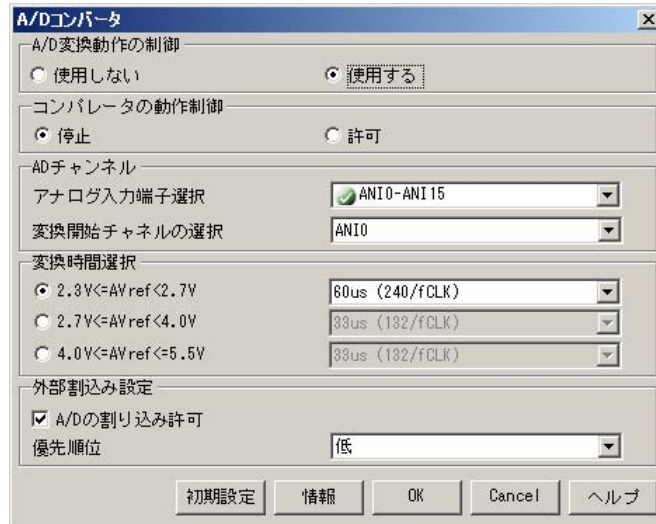
現在設定中の周辺機能を使用するために必要な、他周辺機能の設定および情報を表示します。

6.1.3 設定ダイアログ例

(1) タブがない場合の設定ダイアログ例 (A/D コンバータ)

タブが存在しないときの設定ダイアログ例を次に示します。

図 6 - 10 設定ダイアログ例 (A/D コンバータ)

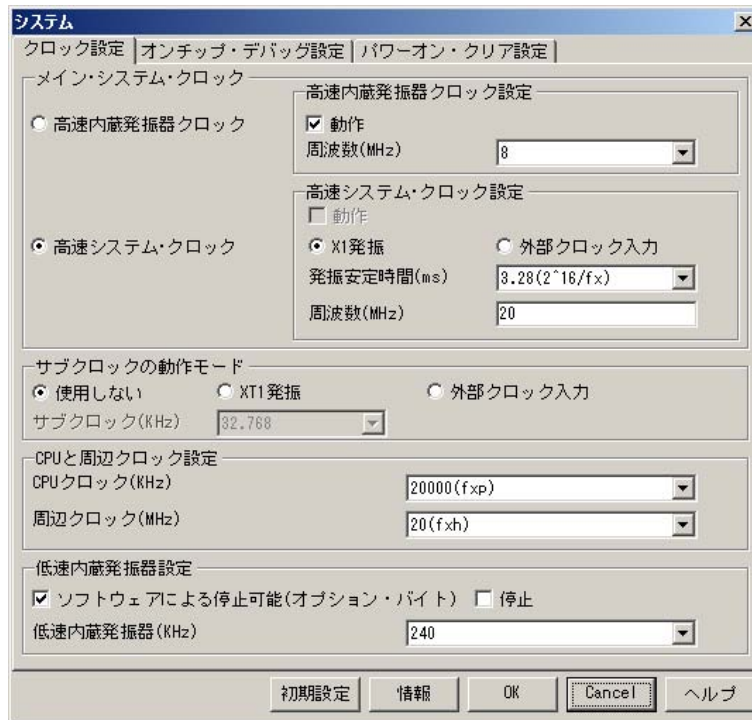


- | | |
|--------|--|
| 初期設定 | : ダイアログの全項目の設定を Applilet2 の初期状態に戻します。 |
| 情報 | : 「詳細情報」ダイアログを表示します。 |
| OK | : 現在の設定内容を確定し設定ダイアログを閉じます。 |
| Cancel | : 現在の設定内容を確定せずに設定ダイアログを閉じます。 |
| ヘルプ | : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。 |

(2) タブがある場合の設定ダイアログ例 (システム系)

タブをクリックして設定する項目を切り替えます。

図 6 - 11 設定ダイアログ例 (システム系)



- | | |
|--------|--|
| 初期設定 | : すべてのタブの全項目の設定を Applilet2 の初期状態に戻します。 |
| 情報 | : すべてのタブに関する内容の「詳細情報」ダイアログを表示します。 |
| OK | : 現在の設定内容を確定し設定ダイアログを閉じます。すべてのタブについて設定内容を確定します。 |
| Cancel | : 現在の設定内容を確定せずに設定ダイアログを閉じます。すべてのタブについて行った設定の変更を無効にします。 |
| ヘルプ | : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。 |

(3) タブがある場合の設定ダイアログ例 (ポート系)

タブをクリックして設定する項目を切り替えます。

設定が済んでいるタブには、チェック記号 を表示します。

図 6 - 12 設定ダイアログ例 (ポート系)



- | | |
|--------|--|
| 初期設定 | : 現在表示中のタブの全項目の設定を Applilet2 の初期状態に戻します。 |
| 情報 | : すべてのタブに関する内容の「詳細情報」ダイアログを表示します。 |
| OK | : 現在の設定内容を確定し設定ダイアログを閉じます。すべてのタブについて設定内容を確定します。 |
| Cancel | : 現在の設定内容を確定せずに設定ダイアログを閉じます。すべてのタブについて行った設定の変更を無効にします。 |
| ヘルプ | : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。 |

(4) タブがある場合の設定ダイアログ例 (D/A, CAN 系)

チャンネルごとにタブがあります。チャンネル・タブ内にさらにタブが存在する場合もあります。タブをクリックして設定する項目を切り替えます。

図 6 - 13 設定ダイアログ例 (D/A 系, 下位のタブが存在しない例)

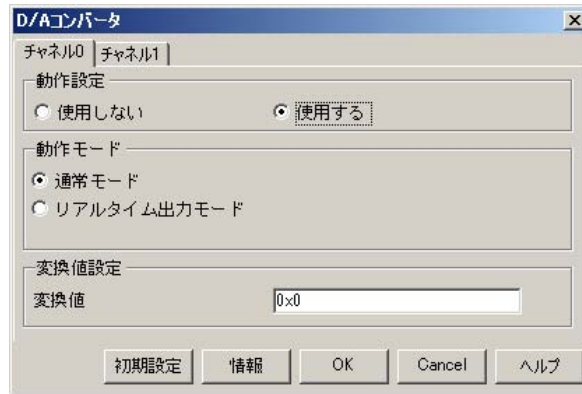
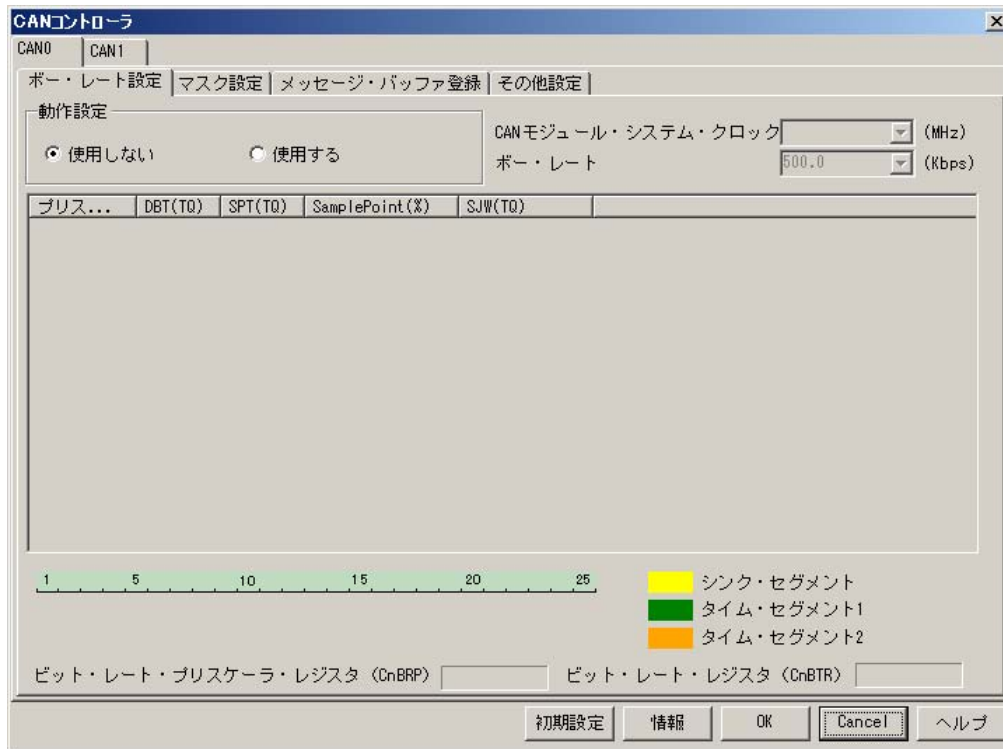


図 6 - 14 設定ダイアログ例 (CAN 系, 下位のタブが存在する例)



- 初期設定** : 現在表示中のタブの全項目の設定を Applilet2 の初期状態に戻します。
- 情報** : すべてのタブに関する内容の「詳細情報」ダイアログを表示します。
- OK** : 現在の設定内容を確定し設定ダイアログを閉じます。すべてのタブについて設定内容を確定します。
- Cancel** : 現在の設定内容を確定せずに設定ダイアログを閉じます。すべてのタブについて行った設定の変更を無効にします。
- ヘルプ** : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。

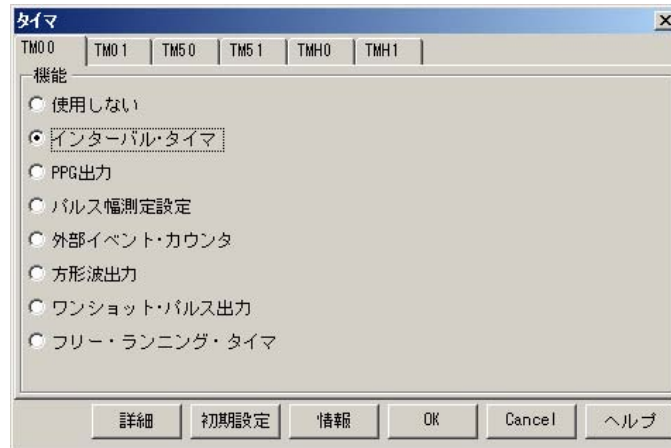
(5) サブ・ダイアログがある場合の設定ダイアログ例 (タイマ, シリアル系)

周辺機能の種類ごとにタブがあり、さらに機能またはモードごとにサブ・ダイアログがあります。

次の手順で、機能を設定します。

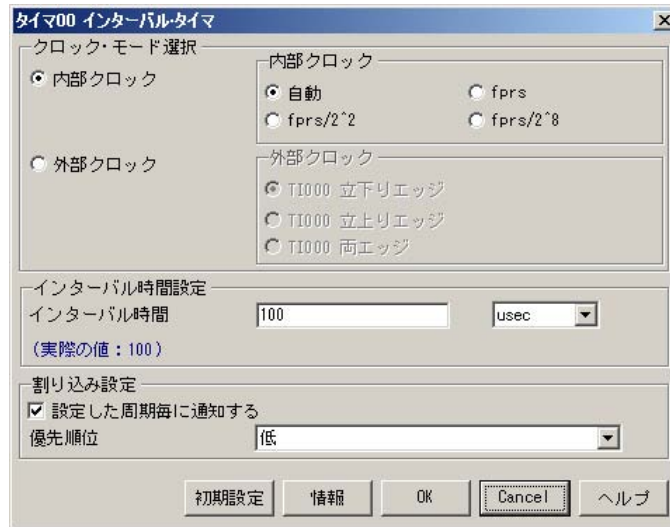
1. タブをクリックして、周辺機能の種類を選択します。
2. 機能またはモードを選択します。
3. [詳細] ボタンを押して、サブ・ダイアログを表示します。
4. サブ・ダイアログで機能を設定します。

図 6 - 15 設定ダイアログ例 (タイマ, シリアル系)



- | | |
|--------|--|
| 詳細 | : 現在表示中のタブで選択している機能 / モードの詳細を設定するサブ・ダイアログを表示します。 |
| 初期設定 | : 現在表示中のタブの機能 / モードの選択、および各機能 / モードのサブ・ダイアログの設定を Applilet2 の初期状態に戻します。 |
| 情報 | : すべてのタブに関する内容の「詳細情報」ダイアログを表示します。 |
| OK | : 現在の設定内容を確定し設定ダイアログを閉じます。すべてのタブについて設定内容を確定します。 |
| Cancel | : 現在の設定内容を確定せずに設定ダイアログを閉じます。すべてのタブについて行った設定の変更を無効にします。 |
| ヘルプ | : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。 |

図 6 - 16 サブ・ダイアログ例 (インターバル・タイマ)



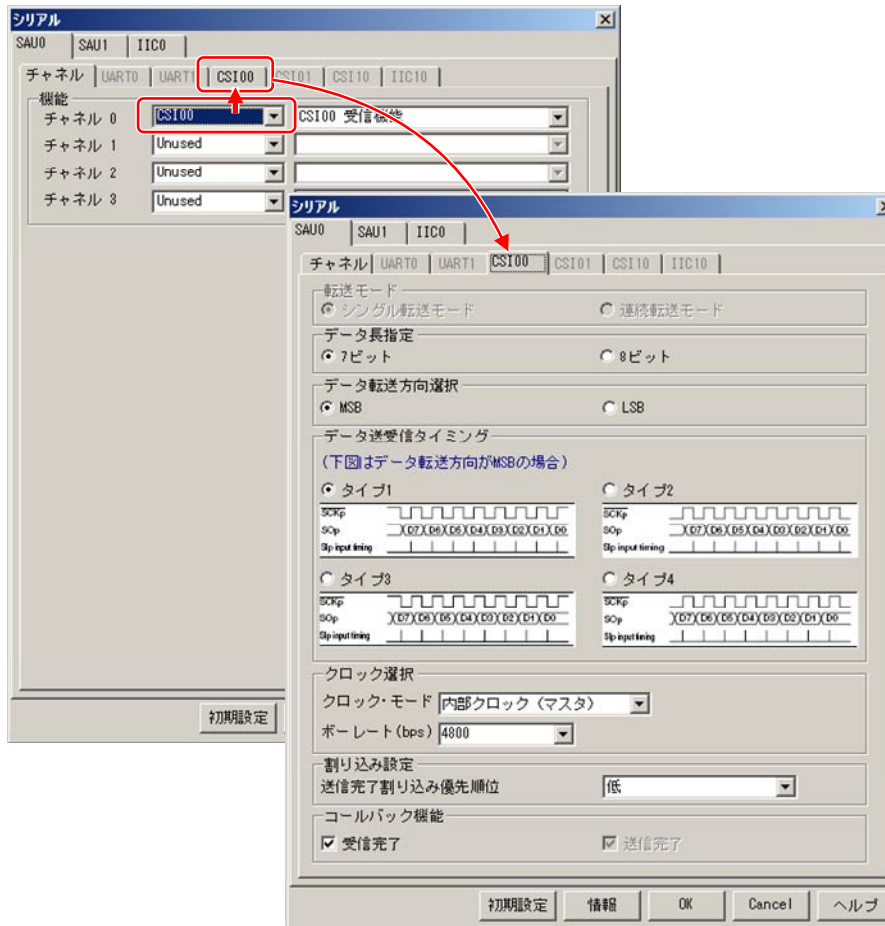
- | | |
|--------|--|
| 初期設定 | : 現在表示中のサブ・ダイアログの設定を Applilet2 の初期状態に戻します。 |
| 情報 | : 現在表示中のサブ・ダイアログに関する内容の「詳細情報」ダイアログを表示します。 |
| OK | : 現在表示中のサブ・ダイアログの設定内容を確定しサブ・ダイアログを閉じます。 |
| Cancel | : 現在表示中のサブ・ダイアログの設定内容を確定せずにサブ・ダイアログを閉じます。 |
| ヘルプ | : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。 |

(6) タブがある場合の設定ダイアログ例 (TAU, SAU 系)

[チャンネル] タブでチャンネルごとに使用する機能を選択します。

機能を選択することにより、その機能 / チャンネルのタブが選択可能になります。

図 6 - 17 設定ダイアログ例 (TAU, SAU 系)



- 初期設定** : 現在表示中のタブ(下位タブを含む)の全項目の設定を Applilet2 の初期状態に戻します。ただし、選択可能 / 不可が変化するタブを表示中の場合、その機能 / チャンネルの設定のみを初期状態に戻します。
- 情報** : 現在表示中のタブに関する内容の「詳細情報」ダイアログを表示します。
- OK** : 現在の設定内容を確定し設定ダイアログを閉じます。すべてのタブについて設定内容を確定します。
- Cancel** : 現在の設定内容を確定せずに設定ダイアログを閉じます。すべてのタブについて行った設定の変更を無効にします。
- ヘルプ** : オンライン・ヘルプを表示します。
現在の設定ダイアログと関連する箇所を表示します。

6.2 「詳細情報」ダイアログ

各周辺機能設定ダイアログで [情報] ボタンを押すと「詳細情報」ダイアログを表示します。「詳細情報」ダイアログには、周辺機能が使用するターゲット・デバイスの資源を表示します。

6.2.1 [I/O レジスタ] タブ

[I/O レジスタ] タブには、周辺機能を使用するために設定する I/O レジスタを表示します。項目名をクリックしても並び替えは行いません。

図 6 - 18 「詳細情報」ダイアログ (I/O レジスタ)

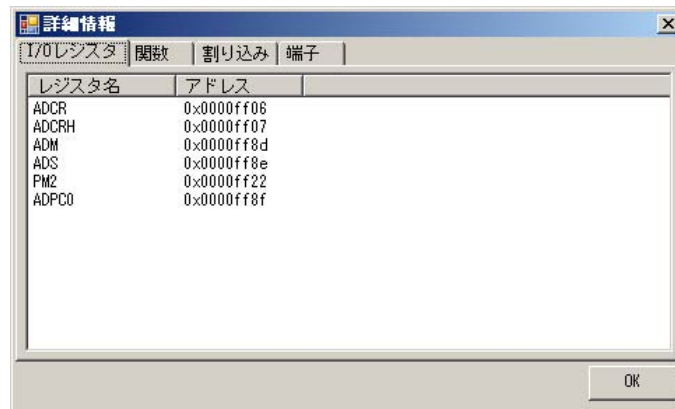


表 6 - 1 「詳細情報」ダイアログの表示項目 (I/O レジスタ)

項目	内容
レジスタ名	I/O レジスタ名
アドレス	I/O レジスタのアドレス

6.2.2 [関数] タブ

[関数] タブには、周辺機能で使用する API 関数を表示します。項目名をクリックしても並び替えは行いません。

図 6 - 19 「詳細情報」ダイアログ (関数)



表 6 - 2 「詳細情報」ダイアログの表示項目 (関数)

項目	内容
関数名	現在の API 関数名 (リネームした場合、リネーム後の API 関数名を表示)
初期設定	API 関数のデフォルト名
ステータス	API 関数の使用状況 “使用されます”: コード生成時に出力します。 “使用しない”: コード生成時に出力しません。
情報	API 関数の説明

6.2.3 [割り込み] タブ

[割り込み] タブには、周辺機能が使用する割り込みを表示します。項目名をクリックしても並び替えは行いません。

図 6 - 20 「詳細情報」ダイアログ (割り込み)

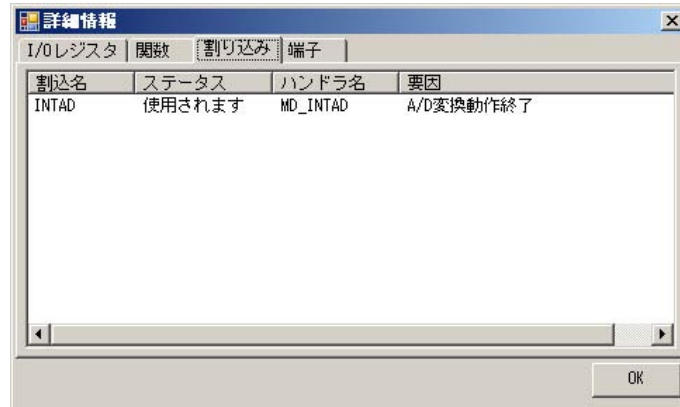


表 6 - 3 「詳細情報」ダイアログの表示項目 (割り込み)

項 目	内 容
割込名	割り込み要因名
ステータス	割り込みの使用状況 “使用されます”: コード生成時に出力します。 “使用しない”: コード生成時に出力しません。
ハンドラ名	割り込み関数名 (リネームした場合, リネーム後の割り込み関数名を表示)
要因	割り込み要因トリガの説明

6.2.4 [端子] タブ

[端子] タブには、周辺機能が使用する端子を表示します。項目名をクリックしても並び替えは行いません。

図 6 - 21 「詳細情報」ダイアログ (端子)

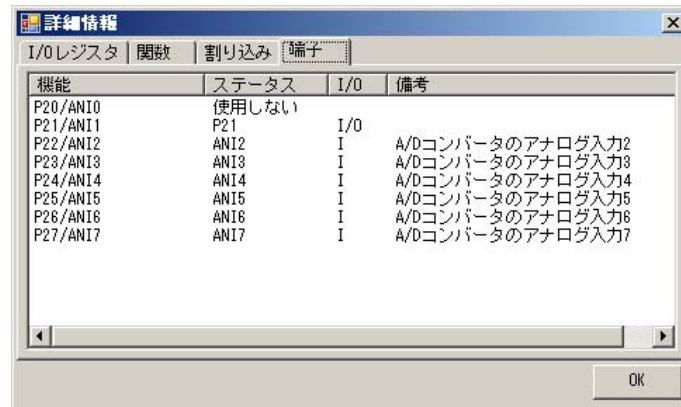


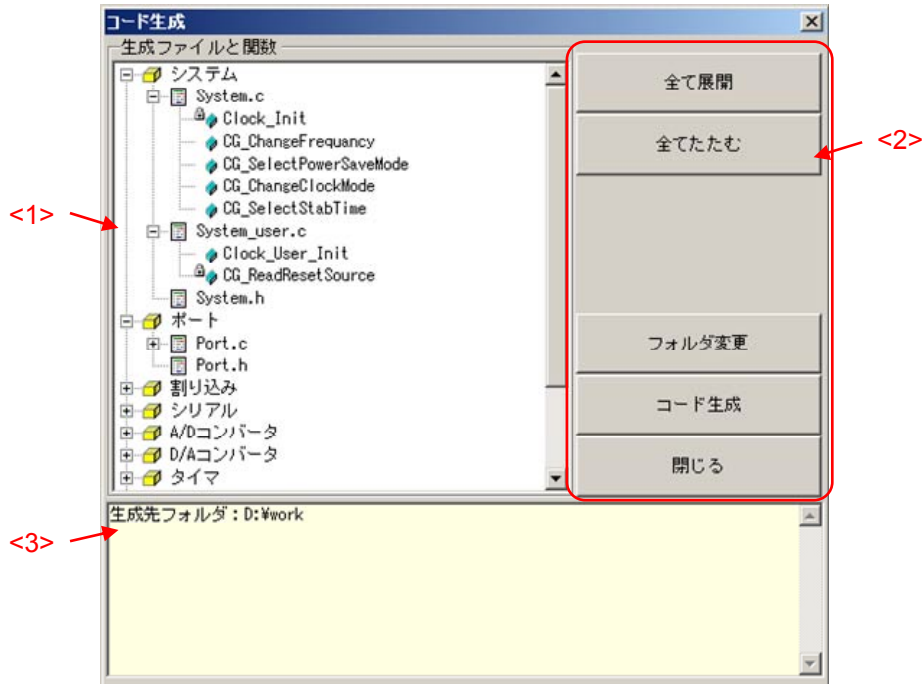
表 6 - 4 「詳細情報」ダイアログの表示項目 (端子)

項目	内容
機能	端子の兼用機能
ステータス	端子の使用状況 “(端子の機能名)”: 表示の機能で使します。 “使用しない”: どの周辺機能も使用していません。
I/O	端子の入出力 “ I ”: 入力, “ O ”: 出力, “ I/O ”: 入出力 ポートとして使用する場合, 入力ポート / 出力ポートに関わらず常に “ I/O ”と表示します。
備考	端子の機能説明

6.3 「コード生成」ダイアログ

「コード生成」ダイアログでは、各周辺機能設定ダイアログの設定内容に従ってコードを生成し、コード生成先のフォルダにファイルを出力します。

図 6-22 「コード生成」ダイアログ








- 備考**
- <1> 生成ファイルと関数エリア
 - <2> ボタン
 - <3> 情報エリア

(1) 生成ファイルと関数エリア

周辺機能，ファイル名，API 名をツリー形式で表示します。

参照のみ可能です。ファイル名や関数名の変更，および“出力する”/“出力しない”の変更はできません。

API 名のアイコンは，次の種類があります。

-  : コード生成時に API を出力しません（現在の周辺機能の設定内容で必須ではありません）。
-  : コード生成時に API を出力します（現在の周辺機能の設定内容で必須ではありませんが，ユーザが出力する設定に変更した）。
-  : コード生成時に API を出力します（現在の周辺機能の設定内容で必須です。ユーザが出力しない設定に変更できない）。
- （黄色） : 割り込み関数です。コード生成時に出力しません。
- （桃色） : 割り込み関数です。コード生成時に出力します。

(2) ボタン

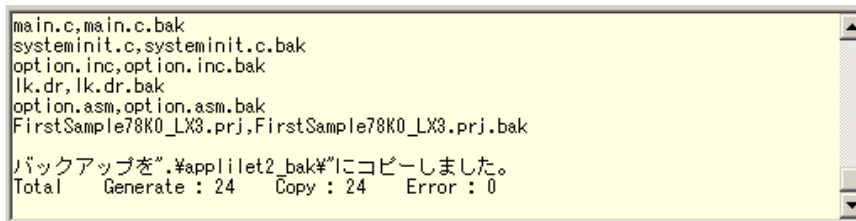
「コード生成」ダイアログには、次のボタンがあります。

全て展開	: 生成ファイルと関数エリアのツリーをすべて展開表示します。
全てたたむ	: 生成ファイルと関数エリアのツリーをすべて閉じて表示します。
フォルダ変更	: コード生成先のフォルダを変更します。ボタンを押すとフォルダ参照ダイアログを表示します。
コード生成	: コード生成を行い、コード生成先のフォルダにファイルを出力します。ボタンを押すと確認ダイアログを表示します。確認ダイアログで、[OK]を押すとコード生成を実行します。[キャンセル]を押すと、コード生成を中止します。
閉じる	: 「コード生成」ダイアログを閉じます。

(3) 情報エリア

ファイルごとにコード生成の状況を表示します。

図 6 - 23 コード生成ダイアログの情報エリア



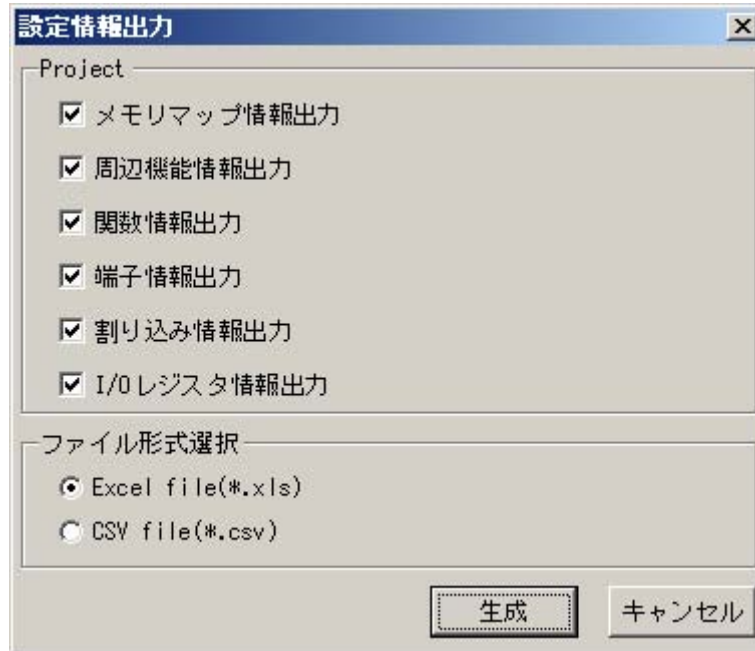
コードを生成すると、最終行に生成したファイルの情報を表示します。

- Generate : 生成したファイルの数
- Copy : 上書きしたファイルの数
- Error : 生成時にエラーが発生したファイルの数

6.4 「設定情報出力」ダイアログ

「設定情報出力」ダイアログでは、Applilet2 の設定情報をファイルに出力します。

図 6 - 24 「設定情報出力」ダイアログ



- Project : 出力する情報を設定します。出力内容は、詳細情報ダイアログの各タブの内容と同等です。
- ファイル形式選択 : 出力するファイルの形式を設定します。エクセル形式または CSV 形式を選択できます。
- 生成** : 設定情報ファイルを出力します。ボタンを押すと「名前をつけて保存」ダイアログを表示します。ファイル名の初期値は“Outputlist”です。
- キャンセル** : ファイル出力をせず、ダイアログを閉じます。

注意 生成の実行後、ファイルの出力が完了するまでに数十秒かかる場合があります。

ファイル出力中は Applilet2 を操作することはできません。完了のメッセージが表示されてから次の操作を行ってください。

第7章 出力ファイル

Applilet2 が生成するファイルは、使用するデバイスおよび選択した環境により異なります。

7.1 V850 系 Applilet2

7.1.1 NEC 環境での出力ファイル

表 7 - 1 V850 系 Applilet2 出力ファイル (NEC 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	crte.s	スタートアップ・ルーチン VSWC レジスタ, OCDM レジスタ, BPC レジスタ設定 ^{注1} , Clock_Init(), BUS_Init() ^{注2} , Systeminit() のコール
	inttab.s	ベクタ設定 リセット・ベクタ, 割り込みベクタ, オプション・バイト ^{注3} , MINICUBE2 用の ROM, RAM 資源の確保 ^{注4}
	systeminit.c	各周辺機能の初期化関数コール ROM 化用コピー関数, CG_ReadResetSource() ^{注5} のコール
	main.c	main 関数
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	850.dir	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注4} , 外部メモリ用セクション ^{注3}
	xxx.prw, xxx.prj	PM+用ワーク・スペース, プロジェクト ROM 化オプション, セキュリティ ID オプション ^{注6}

- 注 1. CAN 内蔵のデバイスのみ出力します。
2. 外部バス・インタフェース使用時のみ出力します。
3. オプション・バイトがあるデバイスのみ出力します。
4. MINICUBE2 使用時のみ出力します。
5. CG_ReadResetSource() の出力を有効にしたときのみ出力します。
6. セキュリティ ID 使用時のみ出力します。

7.1.2 IAR 環境での出力ファイル

表 7 - 2 V850 系 Applilet2 出力ファイル (IAR 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	option.s85 注1	オプション・バイト注2, セキュリティ ID 注3
	systeminit.c	各周辺機能の初期化関数コール, IAR 提供の__low_level_init 関数 __low_level_init 内で VSWC レジスタ, OCDM レジスタ, BPC レジスタ設定 注4, Clock_Init(), BUS_Init()注5, Systeminit() のコール。Systeminit()内で CG_ReadResetSource()注6のコール
	main.c	main 関数
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_Inkxxx.xcl 注7	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション注8, 外部メモリ用セクション注5
	xxx.eww, xxx.ewp	IAR Embedded Workbench 用ワーク・スペース, プロジェクト

注 1. オプション・バイトがあるデバイス, またはセキュリティ ID 使用時に出力します。

2. オプション・バイトがあるデバイスのみ出力します。

3. セキュリティ ID 使用時のみ出力します。

4. CAN 内蔵のデバイスのみ出力します。

5. 外部バス・インタフェース使用時のみ出力します。

6. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

7. MINICUBE2 未使用かつ外部バス・インタフェース未使用時は出力しません。

8. MINICUBE2 使用時のみ出力します。

7.1.3 GHS 環境での出力ファイル

表 7 - 3 V850 系 Applilet2 出力ファイル (GHS 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	startup.s	スタートアップ・ルーチン VSWC レジスタ, OCDM レジスタ, BPC レジスタ設定 ^{注1} , Clock_Init(), BUS_Init() ^{注2} , Systeminit() のコール
	inttab.s	ベクタ設定 リセット・ベクタ, 割り込みベクタ, オプション・バイト ^{注3} , セキュリティ ID ^{注4}
	systeminit.c	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注5} のコール
	main.c	main 関数
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_inkxxx.xcl	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注6} , 外部メモリ用セクション ^{注4}
	default_xxx.gpj, xxx.gpj	MULTI 用プロジェクト

注 1. CAN 内蔵のデバイスのみ出力します。

2. 外部バス・インタフェース使用時のみ出力します。

3. オプション・バイトがあるデバイスのみ出力します。

4. セキュリティ ID 未使用時は 0xffffffffffffff を設定します。何も設定しないと 0 で充填されデバッグできません。

5. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

6. MINICUBE2 使用時のみ出力します。

7.2 78K0R 系 Applilet2

7.2.1 NEC 環境 C 言語ソースでの出力ファイル

表 7 - 4 78K0R 系 Applilet2 出力ファイル (NEC_C 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	systeminit.c	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.c	main 関数
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	lk.dir	リンク・ディレクティブ 外部メモリ用セクション ^{注2}
	xxx.prw, xxx.prj	PM+用ワーク・スペース, プロジェクト オプション・バイト, ユーザ・オプション・バイト, セキュリティ ID ^{注3} オプション

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. 外部バス・インタフェース使用時のみ出力します。

3. セキュリティ ID 使用時のみ出力します。

7.2.2 NEC 環境 ASM 言語ソースでの出力ファイル

表 7 - 5 78K0R 系 Applilet2 出力ファイル (NEC_ASM 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.asm	初期化関数, API 関数
	xxx_user.asm	割り込み関数, コールバック関数
	xxx.inc	extern 宣言
	xxx_equ.inc	レジスタへの代入値マクロを定義
プロジェクト	inttab.asm	リセット・ベクタ, 割り込みベクタ
	systeminit.asm	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.asm	start 関数, main 関数 start 関数内で SP を設定
	macrodriver.inc	全ソース・ファイルで共通使用するマクロ定義
	user_define.inc	空。ユーザ定義用
	lk.dir	リンク・ディレクティブ 外部メモリ用セクション ^{注2}
	xxx.prw, xxx.prj	PM+用ワーク・スペース, プロジェクト オプション・バイト, ユーザ・オプション・バイト, セキュリティ ID ^{注3} オプション

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. 外部バス・インタフェース使用時のみ出力します。

3. セキュリティ ID 使用時のみ出力します。

7.2.3 IAR 環境 C 言語ソースでの出力ファイル

表 7 - 6 78K0R 系 Applilet2 出力ファイル (IAR_C 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	systeminit.c	各周辺機能の初期化関数コール, IAR 提供の __low_level_init 関数。 __low_level_init 内で Systeminit() をコール。 Systeminit() 内で CG_ReadResetSource() ^{注1} をコール。
	main.c	main 関数。 オプション・バイト, ユーザ・オプション・バイト, セキュリティ ID ^{注2}
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_inkxxx.xcl ^{注3}	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注4} , 外部メモリ用セクション ^{注5}
	xxx.eww, xxx.ewp	IAR Embedded Workbench 用ワーク・スペース, プロジェクト

- 注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。
2. セキュリティ ID 使用時のみ出力します。
3. MINICUBE2 未使用かつ外部バス・インタフェース未使用時は出力しません。
4. MINICUBE2 使用時のみ出力します。
5. 外部バス・インタフェース使用時のみ出力します。

7.2.4 IAR 環境 ASM 言語ソースでの出力ファイル

表 7-7 78K0R 系 Applilet2 出力ファイル (IAR_ASM 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.s26	初期化関数, API 関数
	xxx_user.s26	割り込み関数, コールバック関数
	xxx.inc	extern 宣言
	xxx_equ.inc	レジスタへの代入値マクロを定義
プロジェクト	inttab.s26	リセット・ベクタ, 割り込みベクタ
	systeminit.s26	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.s26	start 関数, main 関数 start 関数内で SP を設定, オプション・バイト, ユーザ・オプション・バイト, セキュリティ ID ^{注2}
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_inkxxx.xcl ^{注3}	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注4} , 外部メモリ用セクション ^{注5}
	xxx.eww, xxx.ewp	IAR Embedded Workbench 用ワーク・スペース, プロジェクト

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. セキュリティ ID 使用時のみ出力します。

3. MINICUBE2 未使用かつ外部バス・インタフェース未使用時は出力しません。

4. MINICUBE2 使用時のみ出力します。

5. 外部バス・インタフェース使用時のみ出力します。

7.3 78K0 系 Applilet2

7.3.1 NEC 環境 C 言語ソースでの出力ファイル

表 7 - 8 78K0 系 Applilet2 出力ファイル (NEC_C 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	option.asm	オプション・バイト, MINICUBE2 用 ROM 確保
	option.inc	オプション・バイトへの設定値マクロを定義
	systeminit.c	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.c	main 関数
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	lk.dir	リンク・ディレクティブ 外部メモリ用セクション ^{注2}
	xxx.prw, xxx.prj	PM+用ワーク・スペース, プロジェクト オンチップ・ディバグ・オプション, セキュリティ ID ^{注3} オプション

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. 外部バス・インタフェース使用時のみ出力します。

3. セキュリティ ID 使用時のみ出力します。

7.3.2 NEC 環境 ASM 言語ソースでの出力ファイル

表 7 - 9 78K0 系 Applilet2 出力ファイル (NEC_ASM 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.asm	初期化関数, API 関数
	xxx_user.asm	割り込み関数, コールバック関数
	xxx.inc	extern 宣言
	xxx_equ.inc	レジスタへの代入値マクロを定義
プロジェクト	option.asm	オプション・バイト, MINICUBE2 用 ROM 確保
	option.inc	オプション・バイトへの設定値マクロを定義
	inttab.asm	リセット・ベクタ, 割り込みベクタ
	systeminit.asm	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.asm	start 関数, main 関数 start 関数内で SP を設定
	macrodriver.inc	全ソース・ファイルで共通使用するマクロ定義
	user_define.inc	空。ユーザ定義用
	lk.dir	リンク・ディレクティブ 外部メモリ用セクション ^{注2}
	xxx.prw, xxx.prj	PM+用ワーク・スペース, プロジェクト オンチップ・ディバグ・オプション, セキュリティ ID ^{注3} オプション

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. 外部バス・インタフェース使用時のみ出力します。

3. セキュリティ ID 使用時のみ出力します。

7.3.3 IAR 環境 C 言語ソースでの出力ファイル

表 7 - 10 78K0 系 Applilet2 出力ファイル (IAR_C 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.c	初期化関数, API 関数
	xxx_user.c	割り込み関数, コールバック関数
	xxx.h	レジスタへの代入値マクロを定義
プロジェクト	systeminit.c	各周辺機能の初期化関数コール, IAR 提供の __low_level_init 関数 __low_level_init 内で Systeminit() をコール。 Systeminit() 内で CG_ReadResetSource() ^{注1} のコール
	main.c	main 関数 オプション・バイト, セキュリティ ID ^{注2}
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_inkxxx.xcl ^{注3}	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注4} , 外部メモリ用セクション ^{注5}
	xxx.eww, xxx.ewp	IAR Embedded Workbench 用ワーク・スペース, プロジェクト

- 注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。
2. セキュリティ ID 使用時のみ出力します。
3. MINICUBE2 未使用かつ外部バス・インタフェース未使用時は出力しません。
4. MINICUBE2 使用時のみ出力します。
5. 外部バス・インタフェース使用時のみ出力します。

7.3.4 IAR 環境 ASM 言語ソースでの出力ファイル

表 7 - 11 78K0 系 Applilet2 出力ファイル (IAR_ASM 環境)

出力単位	ファイル名	出力内容
各周辺機能	xxx.s26	初期化関数, API 関数
	xxx_user.s26	割り込み関数, コールバック関数
	xxx.inc	extern 宣言
	xxx_equ.inc	レジスタへの代入値マクロを定義
プロジェクト	inttab.s26	リセット・ベクタ, 割り込みベクタ
	systeminit.s26	各周辺機能の初期化関数コール CG_ReadResetSource() ^{注1} のコール
	main.s26	start 関数, main 関数 start 関数内で SP を設定, オプション・バイト, セキュリティ ID ^{注2}
	macrodriver.h	全ソース・ファイルで共通使用するマクロ定義
	user_define.h	空。ユーザ定義用
	md_inkxxx.xcl ^{注3}	リンク・ディレクティブ MINICUBE2 用の ROM, RAM セクション ^{注4} , 外部メモリ用セクション ^{注5}
	xxx.eww, xxx.ewp	IAR Embedded Workbench 用ワーク・スペース, プロジェクト

注 1. CG_ReadResetSource() の出力を有効にしたときのみ出力します。

2. セキュリティ ID 使用時のみ出力します。

3. MINICUBE2 未使用かつ外部バス・インタフェース未使用時は出力しません。

4. MINICUBE2 使用時のみ出力します。

5. 外部バス・インタフェース使用時のみ出力します。

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。
