

統合開発環境 e² studio

e² studio から EGit を利用する方法

はじめに

EGit は、Eclipse からバージョン管理システム Git を使うためのプラグインです。Git は、プログラムのソース・コードなどの変更履歴を記録・追跡するための分散型バージョン管理システムです。Git では、各ユーザーのワーキング・ディレクトリに、全履歴を含んだリポジトリの完全な複製が作られます。したがって、ネットワークにアクセスできないなどの理由でサーバーのリポジトリにアクセスできない環境でも、履歴の調査や変更の記録といったほとんどの作業を行うことができます。

このドキュメントでは、e² studio 上で EGit を使用方法について説明します。

目次

1. 概要	2
1.1 e ² studio と EGit	2
1.2 動作確認環境	2
2. セットアップ	3
2.1 EGit をインストールする	3
2.2 ユーザー識別情報を設定する	5
3. リポジトリ作成	6
3.1 リモート・リポジトリを作成する	6
3.2 リポジトリのクローンを作成する	7
4. プロジェクトのバージョン管理	10
4.1 リポジトリにプロジェクトを追加する	10
4.2 変更をコミットする	14
4.3 変更箇所をレビューする	15
4.4 バージョン管理からファイルを除外する	16
5. ユーザー間の作業の同期	18
5.1 リモート・リポジトリからリベースする	18
6. 参考情報	23
6.1 Web サイト	23
改訂記録	24

1. 概要

1.1 e² studio と EGit

e² studio は、Eclipse をベースとした統合開発環境です。様々なオープン・ソース・ソフトウェアのプラグインを組み込んで、機能を追加／拡張することができます。

EGit は、Eclipse 上でバージョン管理システム Git を使うためのプラグインです。なお、EGit は Java の Git 実装である JGit を使って動作しますので、別途 Git のコマンドライン・ツールをインストールする必要はありません。

このドキュメントでは、e² studio 上で EGit を使用方法について説明します。

1.2 動作確認環境

このドキュメントで説明する操作手順については、弊社にて以下の環境で確認を実施しています。ただし、オープン・ソースのソフトウェアとの連携になりますので、弊社が動作を保証するものではありません。あらかじめご了解の程お願い申し上げます。

[OS]

- OS Windows10 x64 (日本語版)

[ツール]

- e² studio 2021-07
- EGit 5.12.0.202106070339-r

[プロジェクト]

- デバイス RX64M
- ツールチェーン GCC for Renesas RX C/C++ Toolchain v 8.3.0.202102

2. セットアップ

この章では、EGit を e² studio にインストールしてセットアップする手順について説明します。

e² studio で EGit を使用する場合は、e² studio インストーラで下記オプションをチェックしてインストールしてください。

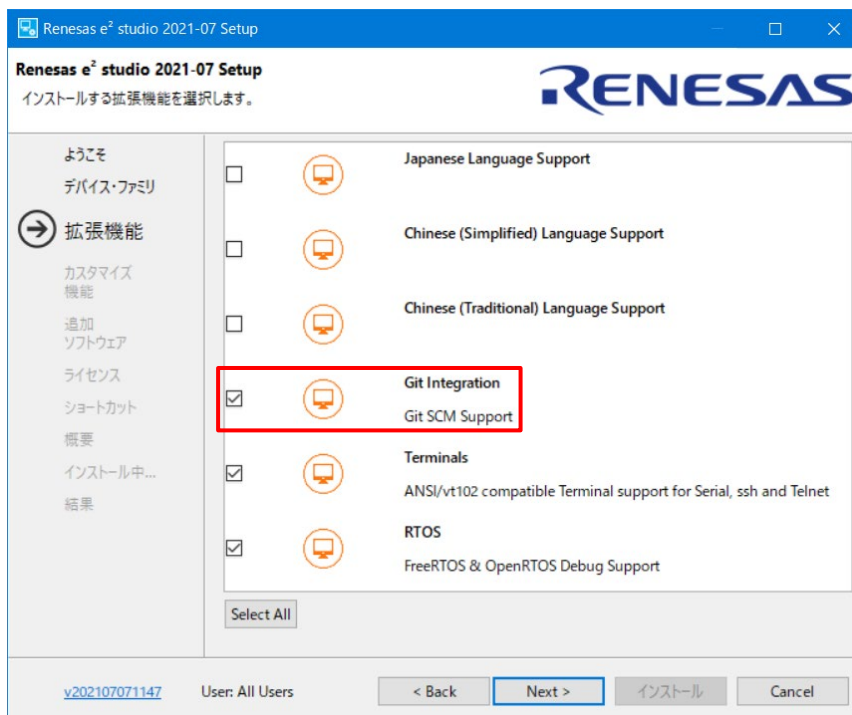


図 1 拡張機能

2.1 EGit をインストールする

EGit は、e² studio に初期状態でインストールされています。最新版を入手したい場合は、アップデート・マネージャーから下記手順に従って EGit 最新バージョンをインストールしてください。

- 1) e² studio を起動し、メニュー [ヘルプ(H)] > [新規ソフトウェアのインストール...] を選択します。
- 2) [インストール - 利用できるソフトウェア] ダイアログが表示されます。[作業対象(W):] 編集ボックスに、EGit のアップデートサイト「<http://download.eclipse.org/egit/updates>」を入力して [追加(A)...] ボタンをクリックします。
- 3) [リポジトリを追加] ダイアログが表示されます。[名前:] 編集ボックスに「EGit updates」（または適切な名前）を入力して、[追加(D)] ボタンをクリックします。

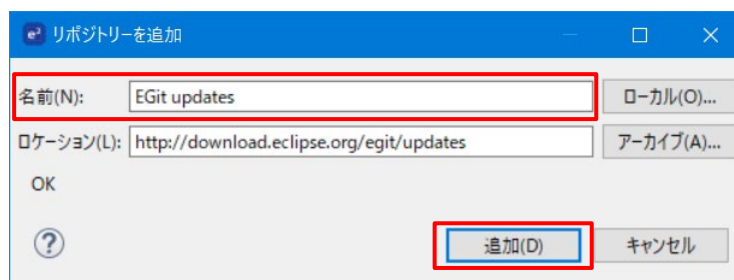


図 2 [リポジトリを追加] ダイアログ

- 4) アップデート・マネージャーが、作業対象を確認して利用できるソフトウェアをリストに表示します。利用できるソフトウェアのリストから「Git integration for Eclipse > Git integration for Eclipse」をチェックして、[次へ(N) >] ボタンをクリックします。

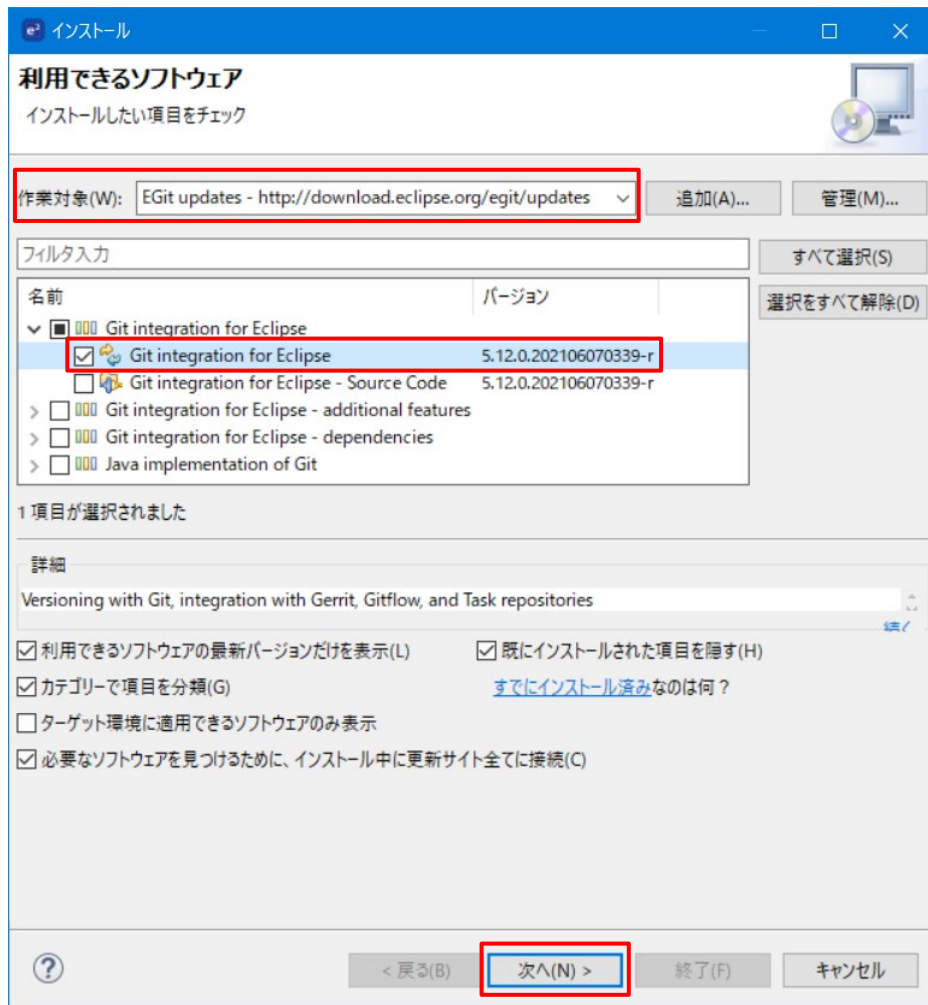


図 3 「インストール - 利用できるソフトウェア」ダイアログ

- 5) 「インストール - インストール詳細」ダイアログが表示されます。[次へ(N) >] ボタンをクリックします。
- 6) 「インストール - ライセンスをレビュー」ダイアログが表示されます。ライセンス・テキストを確認して、[使用条件の条項に同意します(A)] チェックボックスをチェックします。次に [終了(F)] ボタンをクリックします。
- 7) インストールが開始されます（インストールには、時間がかかります）。EGit のインストールが完了すると、e² studio を再起動する必要がある旨の [ソフトウェア更新] メッセージが表示されますので、[すぐに再始動(R)] ボタンをクリックします。

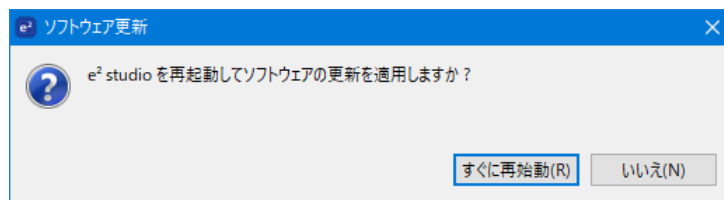


図 4 「ソフトウェア更新」メッセージ

- 8) e² studio が再起動したら、インストール作業は終了です。メニュー [ウィンドウ(W)] > [パースペクティブ(R)] > [パースペクティブを開く(O)] > [その他(O)...] を選択します。[パースペクティブを開く] ダイアログが表示され、「Git」パースペクティブが選択できるようになります。

2.2 ユーザー識別情報を設定する

リポジトリの履歴が変更される時（技術的には、コミットが行われたとき）、Git はそのコミットを行ったユーザーを記録します。ユーザーを識別する情報は、名前と電子メールアドレスです。この情報は、「<user-home-folder> ¥.gitconfig」ファイルに保存されています。

下記手順に従って、ユーザー識別情報を編集します。

- 1) メニュー [ウィンドウ(W)] > [設定(P)] を選択します。
- 2) [設定] ダイアログが表示されます。ダイアログ左側ツリーの「Version Control (Team) > Git > Configuration」を選択して [Configuration] パネルを表示します。
- 3) [User Settings] タブをクリックし、[Add Entry...] ボタンをクリックします。
- 4) [Add a configuration entry] ダイアログが表示されます。以下の情報を入力して、[Add] ボタンをクリックします。
 - Key: 「user.name」を入力します。
 - Value: 名前を入力してください。
- 5) 再度、[Add Entry...] ボタンをクリックします。[Add a configuration entry] ダイアログが表示されますので、下記の e-mail 情報を入力して、[Add] ボタンをクリックします。
 - Key: 「user.email」を入力します。
 - Value: メールアドレスを入力してください。

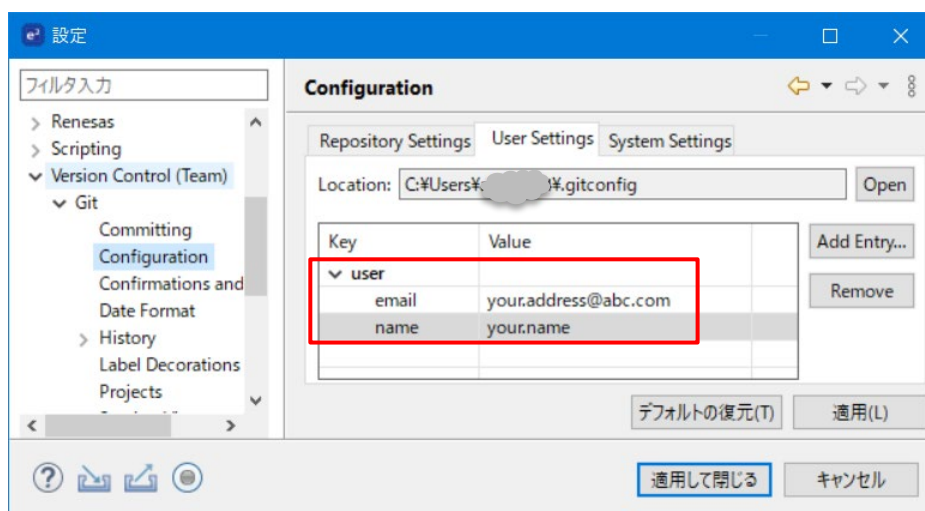


図 5 「設定」ダイアログ

- 6) [適用して閉じる] ボタンをクリックします。設定情報が、[Location] で示された.gitconfig ファイルに保存されます。

3. リポジトリ作成

リポジトリとは、ファイルやディレクトリの状態を記録する場所です。保存された状態は、内容の変更履歴として格納されています。

この章では、新しいリポジトリを作成する方法について説明します。

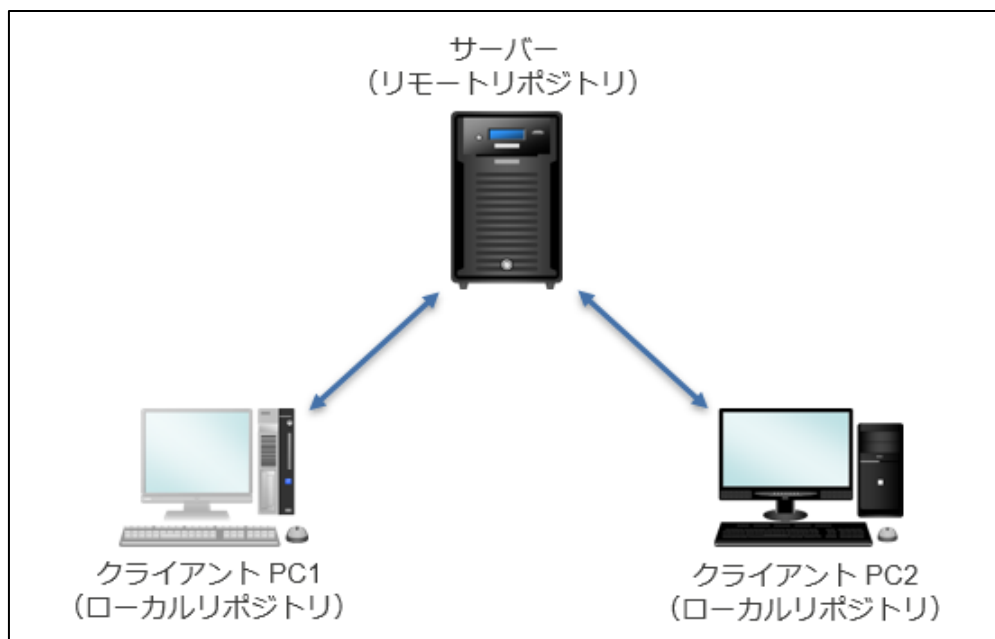


図 6 リポジトリのネットワーク構成

3.1 リモート・リポジトリを作成する

作成するリポジトリをリモート・リポジトリとして使用するには、リポジトリを複数の PC からアクセス可能にする必要があります。そのため、ネットワーク上のフォルダーに作成するのが一般的です。リモート・リポジトリは、GitHub、Amazon CodeCommit などのオンラインサービス上に作成することもできます。

下記手順に従って、ローカルネットワーク上にリモート・リポジトリを作成します。

- 1) e² studio を起動し、メニュー [ウィンドウ(W)] > [パースペクティブ(P)] > [パースペクティブを開く(O)] > [その他(O)...] を選択します。
- 2) [パースペクティブを開く] ダイアログが表示されます。「Git」を選択して、[開く(o)] ボタンをクリックします。
- 3) [Git Repositories] ビューで、[Create a new Git Repository and it to this view] ボタンをクリックします。

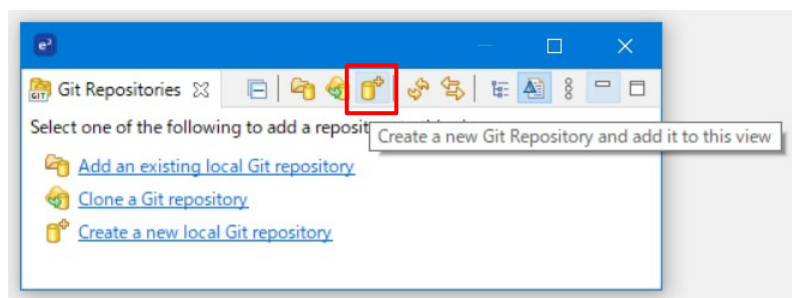


図 7 [Create a new Git Repository and it to this view] ボタン

- 4) [Create a Git Repository] ダイアログが表示されます。[Repository directory:] 編集ボックスにローカル・ネットワーク・フォルダーを入力して、[Create] ボタンをクリックします。

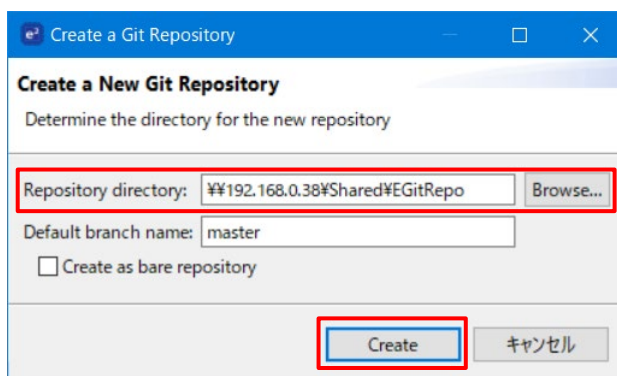


図 8 [Create a Git Repository] ダイアログ

- 5) ユーザーは、フォルダーにアクセスするためのネットワーク資格情報を求められる場合があります。ユーザー名とパスワードを入力し、[資格情報を記憶する] チェックボックスをチェックし、[OK] ボタンをクリックしてください。

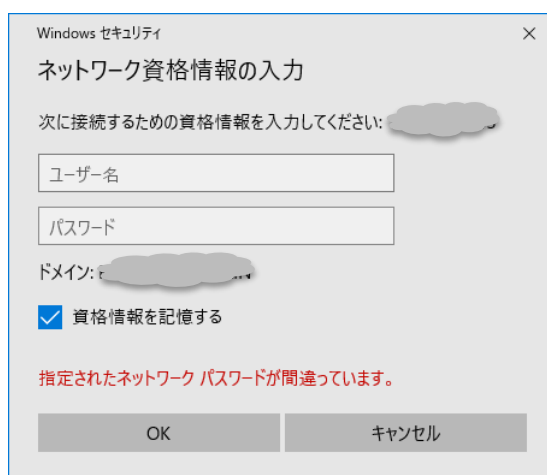


図 9 ネットワーク資格情報の入力

- 6) リモート・リポジトリが作成されます。

3.2 リポジトリのクローンを作成する

作成したリモート・リポジトリは、複数人で共有するためのリポジトリです。ユーザー一人が利用するためには、クライアント PC に、リモート・リポジトリのクローンであるローカル・リポジトリを作成する必要があります。

- 1) 「Git」 パースペクティブを選択します。[Git Repositories] ビューで、[Clone a Git Repository and add the clone to this view] ボタンをクリックします。

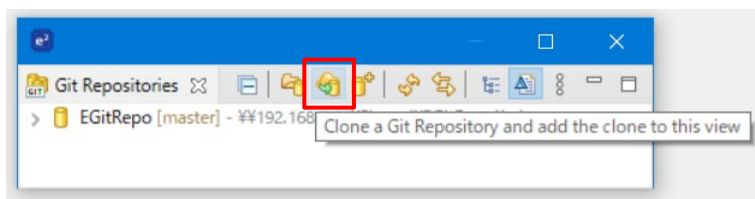


図 10 [Clone a Git Repository and add the clone to this view] ボタン

- 2) [Clone Git Repository - Source Git Repository] ダイアログが表示されます。[URL:] 編集ボックスに前章で作成したリモート・リポジトリのフォルダーを入力して、[次へ(N) >] ボタンをクリックします。

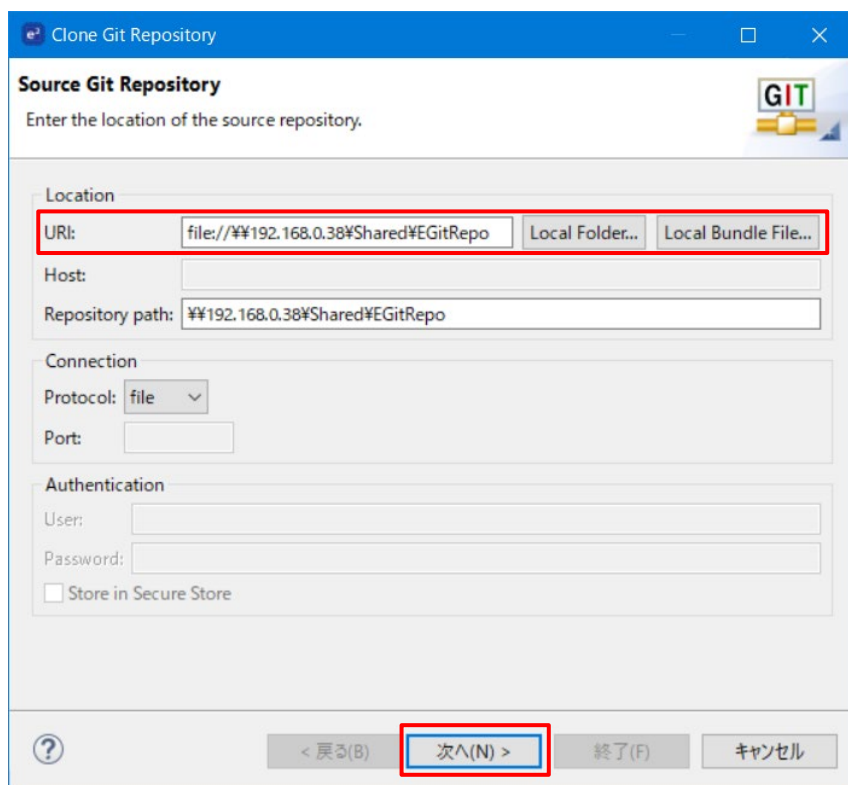


図 11 [Clone Git Repository - Source Git Repository] ダイアログ

- 3) [Clone Git Repository - Branch Selection] ダイアログが表示されます。[次へ(N) >] ボタンをクリックします。(リポジトリが新規の場合、ブランチはありません。新規ではない場合、ブランチを選択できます)

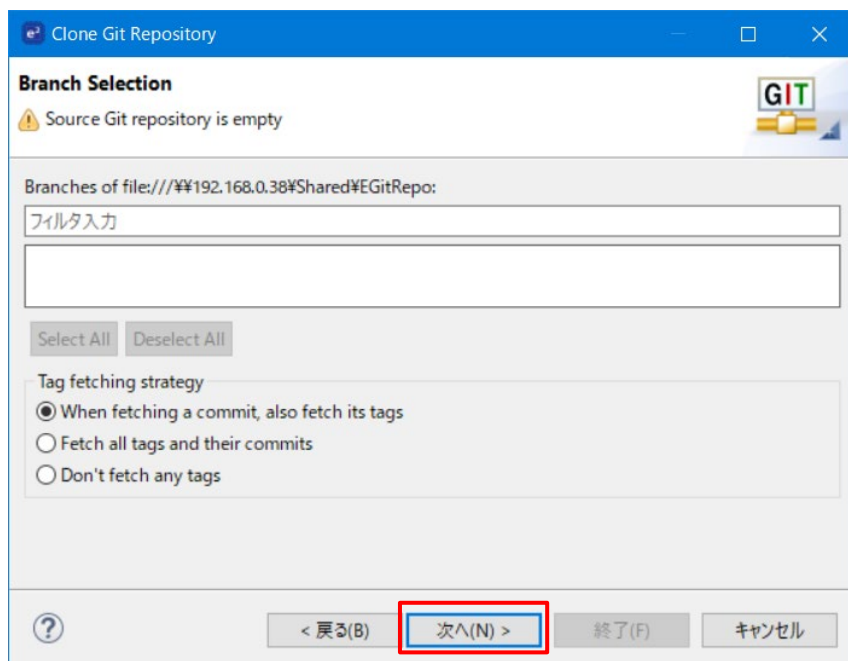


図 12 [Clone Git Repository - Branch Selection] ダイアログ

- 4) [Clone Git Repository - Local Destination] ダイアログが表示されます。[Destination] グループボックスの [Directory:] 編集ボックスにクライアント PC のフォルダーを指定します。ここでは、例として「C:\workspace\EGitClone\clone1」を指定します。次に [終了(F)] ボタンをクリックします。

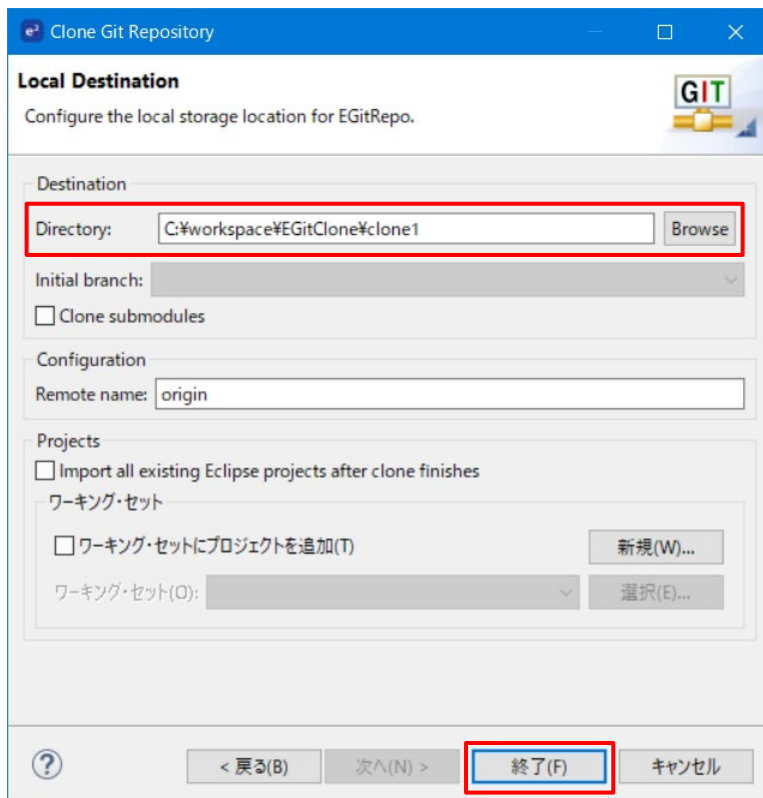


図 13 [Clone Git Repository - Local Destination] ダイアログ

- 5) ローカル・リポジトリが作成されます。

4. プロジェクトのバージョン管理

この章では、リポジトリにプロジェクトを追加して、プロジェクトのバージョン管理を実施する方法について説明します。

バージョンを管理するファイルは、メニュー [ウィンドウ(W)] > [設定(P)] を選択すると表示される [設定] ダイアログの [Version Control (Team)] > [無視するリソース] で指定することができます。

ソース・ファイル、ヘッダ・ファイル、およびピリオド「.」で始まるファイル（.project と .cproject）が管理対象になるよう設定されていなければ問題ないでしょう。

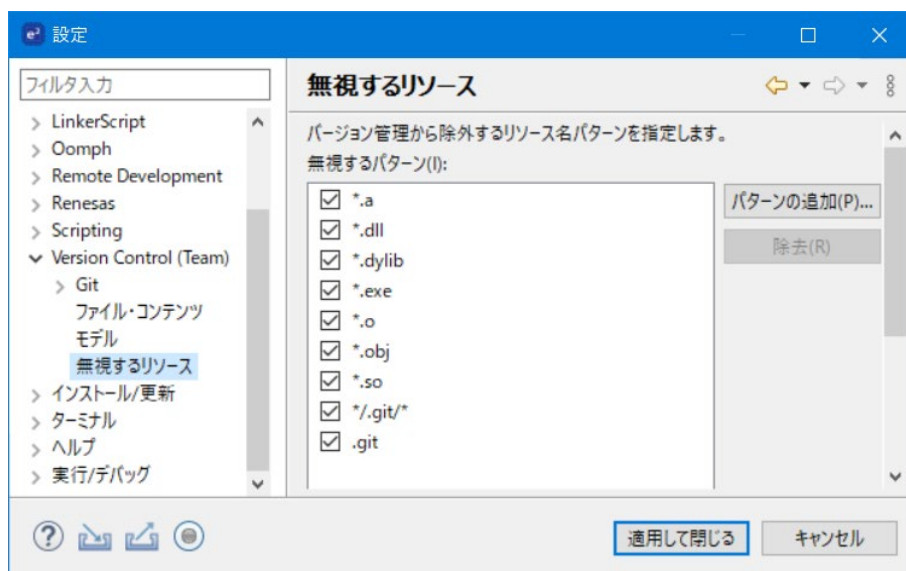


図 14 無視するリソース

4.1 リポジトリにプロジェクトを追加する

GCC for Renesas RX 用の Executable プロジェクトを作成して、そのプロジェクトをリポジトリに追加します。

- 1) 「C/C++」 パースペクティブを選択します。メニュー [ファイル(F)] > [新規(N)] > [C/C++ Project] を選択します。
- 2) [New C/C++ Project - Templates for New C/C++ Project] ダイアログが表示されます。左側リストボックスで「Renesas RX」を選択し、右側リストボックスで「GCC for Renesas RX C/C++ Executable Project」を選択します。[次へ(N)>] ボタンをクリックします。

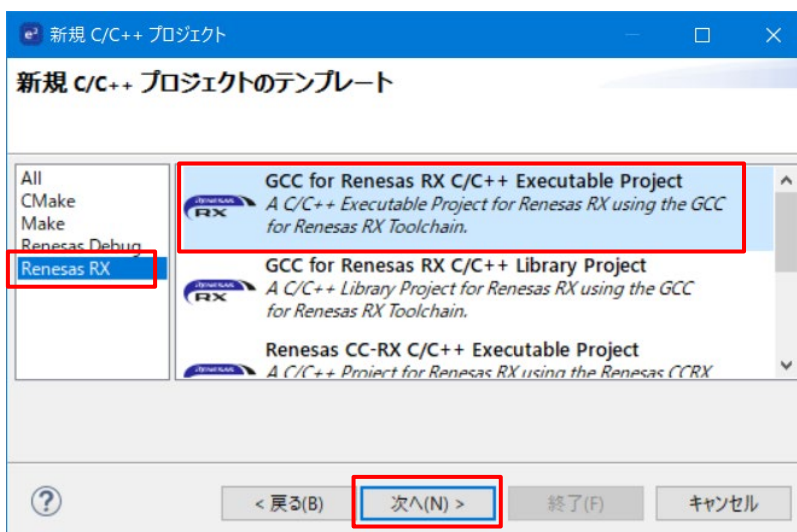


図 15 [New C/C++ Project – Templates for New C/C++ Project] ダイアログ

- 3) [GCC for Renesas RX - New GCC for Renesas RX Executable Project] ダイアログが表示されます。
[プロジェクト名(P):] 編集ボックスに「SampleEGit」を入力して、[次へ(N) >] ボタンをクリックします。
- 4) [GCC for Renesas RX - Select toolchain, device & debug settings] ダイアログが表示されます。下記情報を設定します。残りの箇所は、デフォルトのままとします。
 - [ツールチェーン:] コンボボックス
「GCC for Renesas RX」
 - [ツールチェーン・バージョン] コンボボックス
「8.3.0.202102」およびそれ以降
 - [ターゲット・デバイス] 編集ボックス
例 : RX600 > RX64M > RX64M - 176pin > R5F564MLCxFC

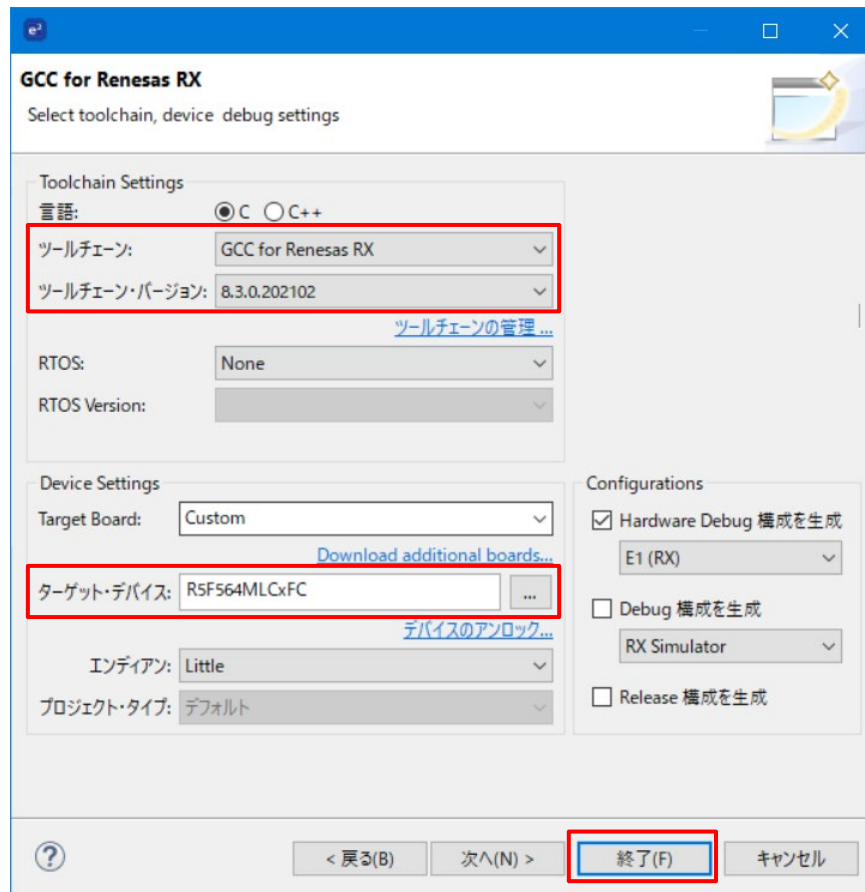


図 16 【GCC for Renesas RX - Select toolchain, device & debug settings】ダイアログ

- 5) [終了(F)] ボタンをクリックします。SampleEGit プロジェクトが作成され、[プロジェクト・エクスプローラー] ビュー上に表示されます。
- 6) ビルドを実行して、プロジェクトに問題がないことを確認します。
- 7) リポジトリにプロジェクトをコミットする前に、クリーンを実行してビルドにより生成されたファイルを削除します。
- 8) [プロジェクト・エクスプローラー] ビューで SampleEGit プロジェクトを選択して、コンテキスト・メニュー [Team] > [プロジェクトの共有(S)...] を選択します。
- 9) [Share Project - Configure Git Repository] ダイアログが表示されます。[Repository:] コンボボックスでローカル・リポジトリを選択し、[終了(F)] ボタンをクリックします。

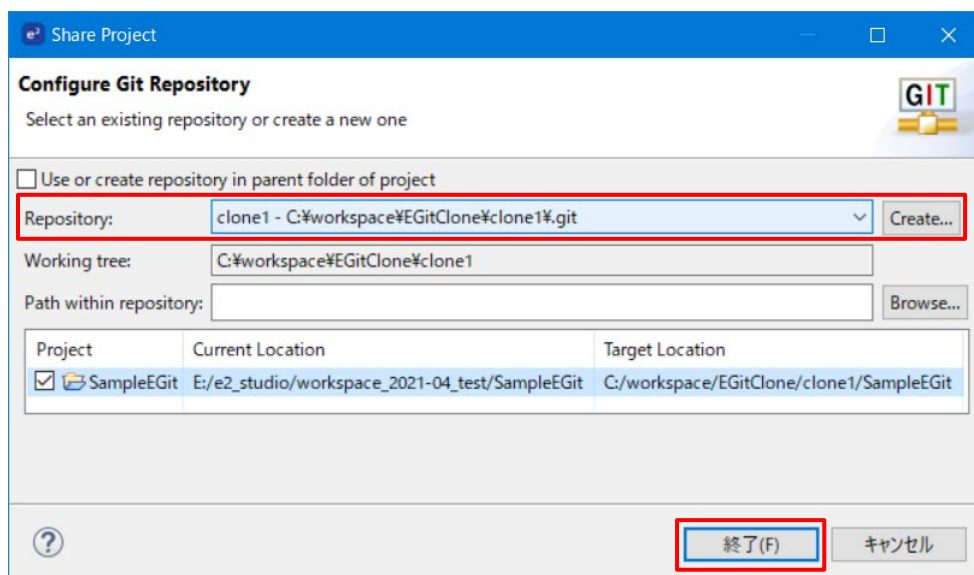


図 17 【Share Project - Configure Git Repository】ダイアログ

- 10) 「Git」 パースペクティブを選択します。[Git Repositories] ビューでローカル・リポジトリを選択します。
- 11) [Git Staging] ビューで、プロジェクト内の新しいファイルが [Unstaged Changes] リストボックスに表示されます。表示されない場合は、[Refresh] ボタンをクリックしてください。

注：[Link with Editor and Selection] ボタンが ON の場合、ステージングされていない変更が自動的に更新されます。

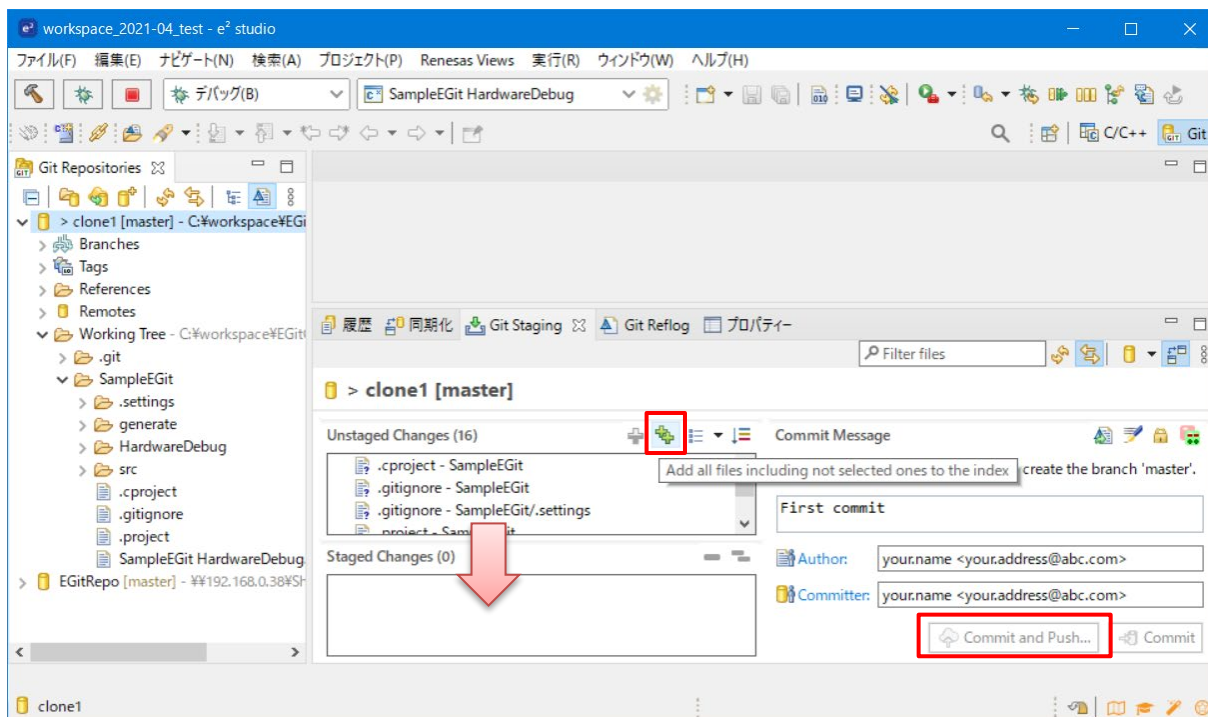


図 18 プロジェクトのコミット

- 12) [Add all files including not selected ones to the index] ボタンをクリックして、[Unstaged Changes] リストボックスのすべてのアイテムを [Staged Changes] リストボックスに移動します。次に、[Commit Message] 編集ボックスに、メッセージを入力します。[Commit and Push...] ボタンをクリックします。

- [Commit and Push...] ボタン：ローカル・リポジトリにコミットし、すぐにリモート・リポジトリにプッシュします。
 - [Commit] ボタン：ローカル・リポジトリのみにコミットする
- 13) [Push Branch master] ダイアログが表示されます。リモート・リポジトリにプッシュするのが初めての場合、リモート・リポジトリの設定とプッシュ動作を指定する必要があります。

[Remote:] コンボボックスで、リモート・リポジトリを選択し、[When pulling:] コンボボックスで「Rebase」を選択します。[Preview >] ボタンをクリックします。

これがリモート・リポジトリへの最初のコミットである場合、このステップで「master」ブランチが作成されます。

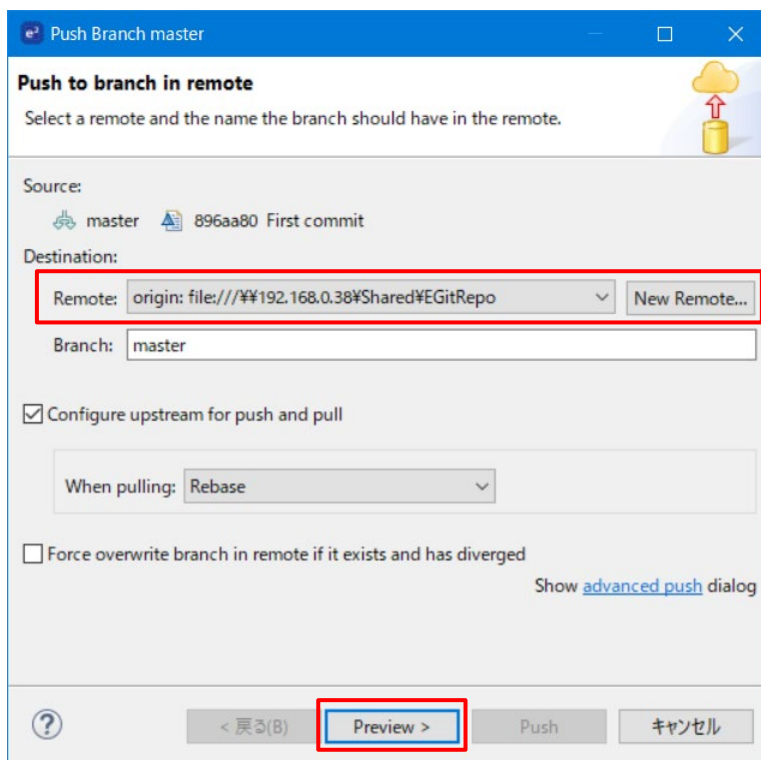


図 19 [Push Branch master] ダイアログ

- 14) [Push Branch master - Push Confirmation] ダイアログが表示されます。[Push] ボタンをクリックします。
- 15) [Push Results] ダイアログが表示されます。メッセージを確認して、[閉じる(C)] ボタンをクリックします。

4.2 変更をコミットする

ソース・ファイルを変更して、リポジトリにコミットします。

- 1) 「C/C++」 パースペクティブを選択します。「SampleEGit.c」ファイルをエディタでオープンします。
- 2) 下記の通りファイルを編集して保存します。

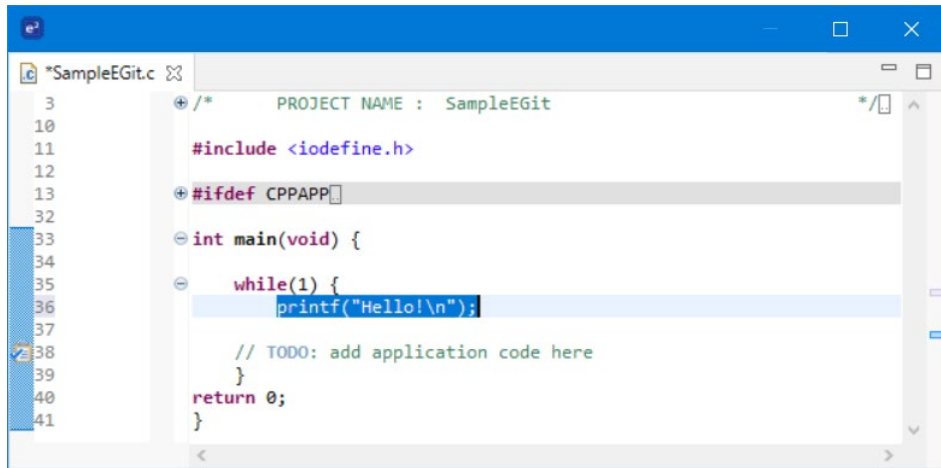


図 20 「SampleEGit.c」ファイル

- 3) 「Git」 パースペクティブを選択します。[Git Repositories] ビューでローカル・リポジトリを選択します。[Git Staging] ビューで、[Unstaged Changes] リストボックスに変更したファイルが表示されます。表示されない場合は、[Refresh] ボタンをクリックしてください。
- 4) [Add all files including not selected ones to the index] ボタンをクリックして、[Unstaged Changes] リストボックスの変更ファイルを [Staged Changes] リストボックスに移動します。次に、[Commit Message] 編集ボックスにメッセージ（例：Update main()）を入力し、[Commit and Push...] ボタンをクリックします。
- 5) [Push Results] ダイアログが表示されます。内容を確認して [閉じる(C)] ボタンをクリックします。

4.3 変更箇所をレビューする

編集したソース・ファイルの変更内容を確認します。

- 1) 「Git」 パースペクティブを選択します。[履歴] ビューを表示し、右下リストの「SampleEGit/src/SampleEGit.c」を選択します。左下にファイルの編集履歴が表示されます。

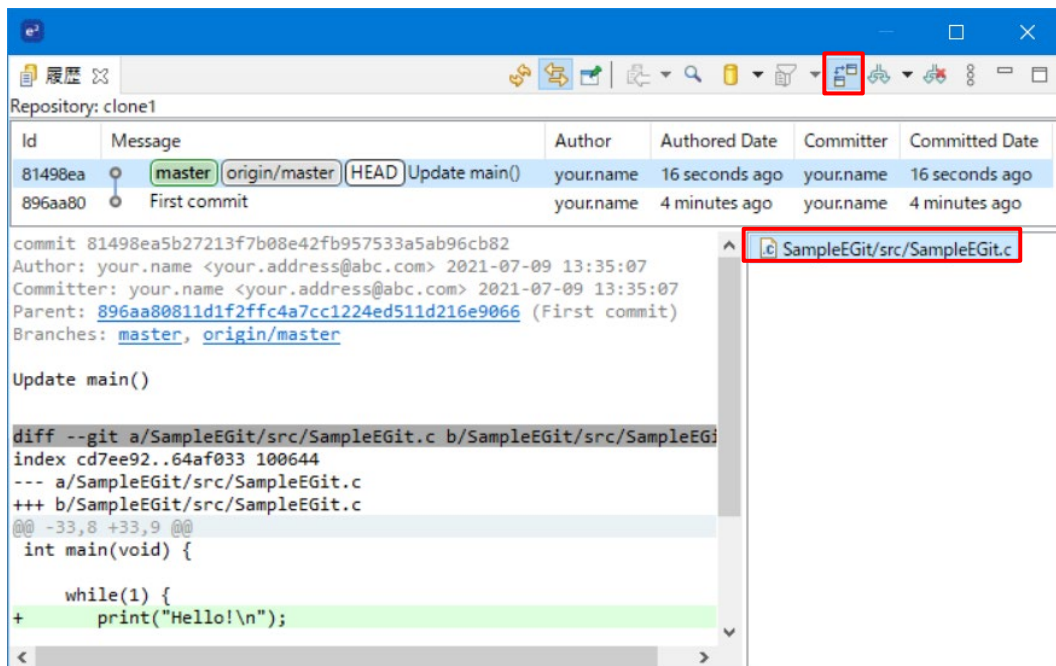


図 21 [履歴] ビュー

- 2) [Compare Mode] ボタンをクリックして ON にします。右下リストの「SampleEGit/src/SampleEGit.c」をダブルクリックすると、[C 比較ビューアー] ビューが表示されます。

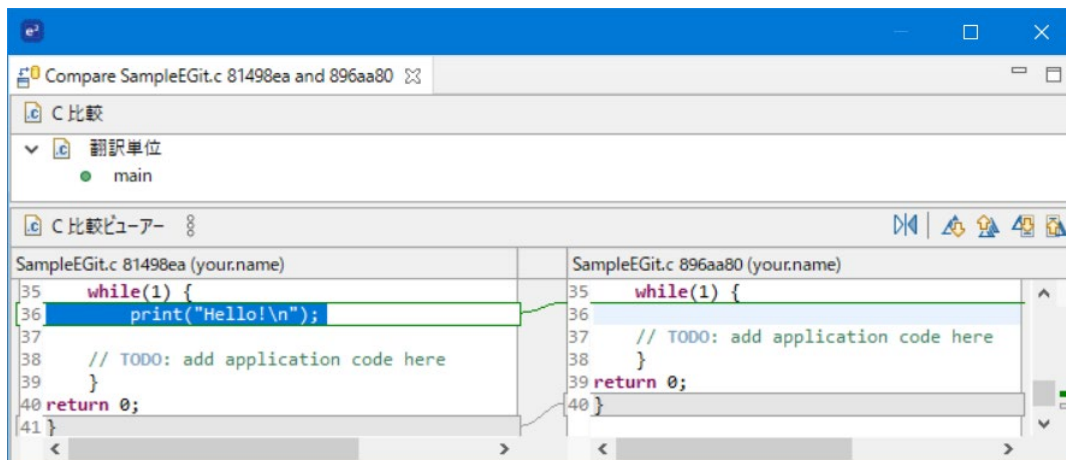


図 22 [C 比較] ビュー

4.4 バージョン管理からファイルを除外する

バージョン管理の対象としないファイルの設定を行います。

- 1) 「C/C++」 パースペクティブを選択します。[プロジェクト・エクスプローラー] でバージョン管理から除外するファイル、またはフォルダーを選択して、コンテキスト・メニュー [Team] > [Ignore] を選択します。
- 2) 「Git」 パースペクティブを選択します。[Git Repositories] ビューで、[更新(F5)] ボタンをクリックして、ローカル・リポジトリの選択したファイルと同じフォルダーを展開します。.gitignore ファイルが作成されており、選択したファイルのパスが記載されています。

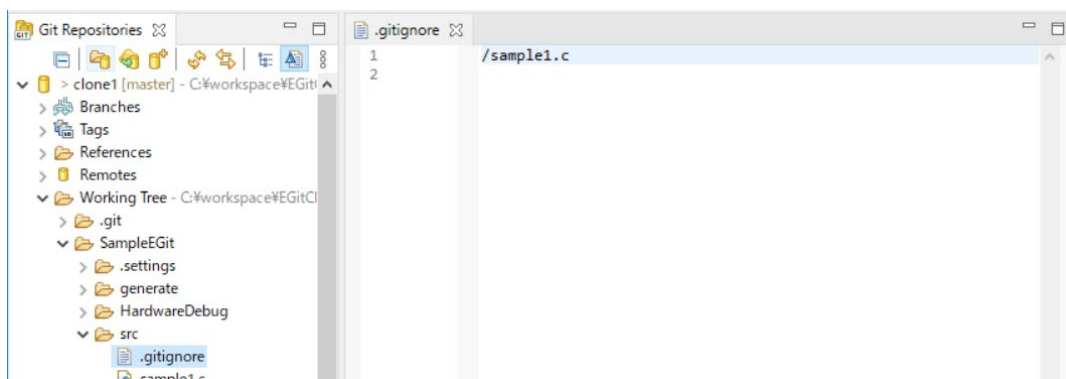


図 23 .gitignore ファイル

- 3) ステップ(1)を実行するたびに、ファイルと同じフォルダーに.gitignore ファイルが（ファイルが存在しない場合）作成されます。フォルダー毎に作成、管理することが煩雑な場合は、プロジェクトフォルダーに.gitignore ファイルを1つ作成して（各フォルダーの.gitignore ファイルは削除する）、バージョン管理から除外するファイルをパス付きで記載してください。フォルダー毎に.gitignore ファイルを作成した場合と同等の結果を得ることができます。

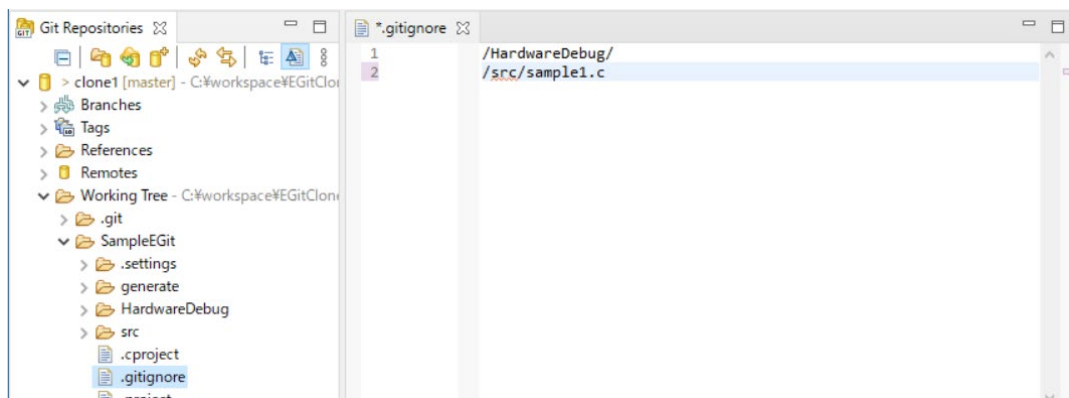


図 24 プロジェクトに 1 つだけの.gitignore ファイル

- 4) .gitignore ファイルは、4.2 章の手順に従ってコミットしてください。

5. ユーザー間の作業の同期

EGit 使用したバージョン管理作業の標準的なワークフローを以下に示します。

- 1) 共通サーバーにリモート・リポジトリを作成します。
- 2) ユーザーA は、リモート・リポジトリのクローンをローカル・リポジトリとして作成し、そのローカル・リポジトリにプロジェクトを追加してバージョン管理を開始します。そして変更をコミット（およびプッシュ）します。
- 3) ユーザーB は、リモート・リポジトリからローカル・リポジトリを作成し、そのローカル・リポジトリを使用して作業を開始します。ユーザーA の更新をリモート・リポジトリから取得するためにフェッチを実施し、ローカルとリモートのリポジトリを同期するためにリベースを実施します（競合がある場合は、それらを解決してリベースを続行します）。そして、自身の変更をリポジトリにコミット&プッシュします。

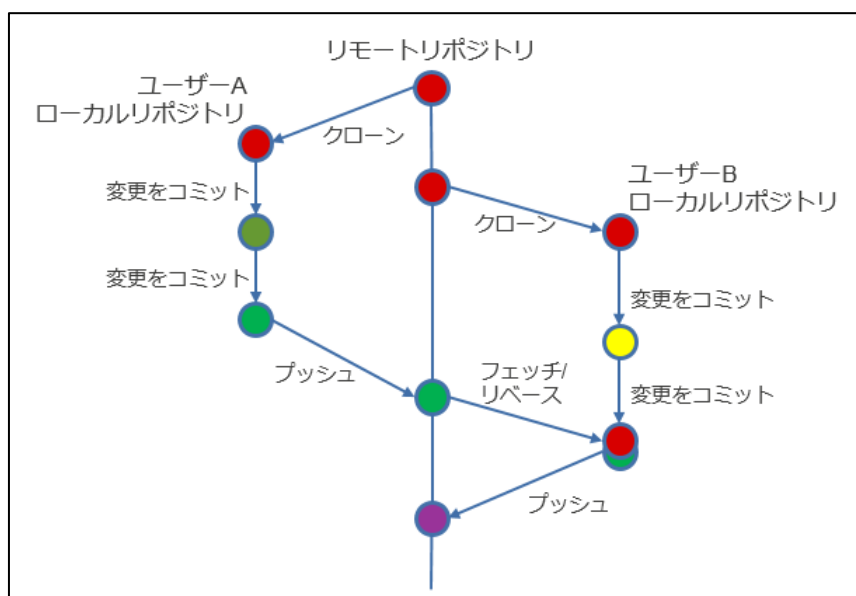


図 25 標準的なワークフロー

この章では、リモート・リポジトリからローカル・リポジトリにリベースする方法について説明します。

5.1 リモート・リポジトリからリベースする

他のユーザーがリモート・リポジトリに新しい情報をプッシュすると、自身のローカル・リポジトリの情報が古くなり、作業を継続するにはリベースを実施する必要があります。リモート・リポジトリに何かをプッシュする場合は、競合を避けるために最初にリベースすることをお勧めします。

下記手順に従って、リベースを実施します。

- 1) 別のクライアント PC を用意します。3.1 章で作成したリモート・リポジトリを使用して、3.2 章の手順に従ってローカル・リポジトリを作成します。

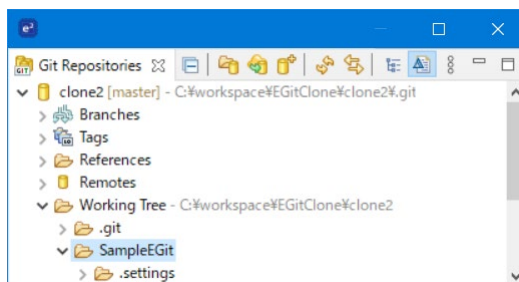


図 26 リモート・リポジトリから作成したローカル・リポジトリ

- 2) [Git Repositories] ビューで、「clone2」の「Working Tree」から SampleEGit.c を開き、それを変更し、コミットして、リモート・リポジトリにプッシュします。「clone2」の「Local」ブランチと「Remote tracking」ブランチが同じであることが分かります。

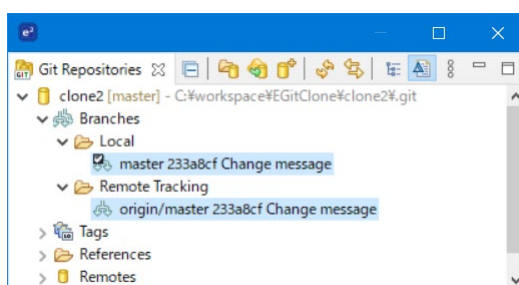


図 27 プッシュ後の「Local」ブランチと「Remote tracking」ブランチ

- 3) 元のクライアント PC に戻り [Git Repositories] ビューで、「clone1」の「Working Tree」から、SampleEGit.c を開き、「clone2」で変更した個所と同じ個所を変更し、コミットして、リモート・リポジトリにプッシュします。競合に関するエラーメッセージが表示されます。「clone1」の「Local」ブランチと「Remote tracking」ブランチが異なっていることが分かります。

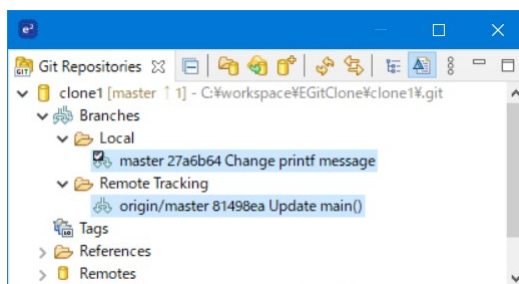


図 28 プッシュ時の競合

- 4) [Git Repositories] ビューで、「clone1」を選択し、コンテキスト・メニュー [Fetch from origin] を選択します。
- 5) [Fetch Results] ダイアログが表示されます。「clone2」によりコミットされたリモート・リポジトリからの情報が表示されます。[Close] ボタンをクリックして、[Fetch Results] ダイアログを閉じます。

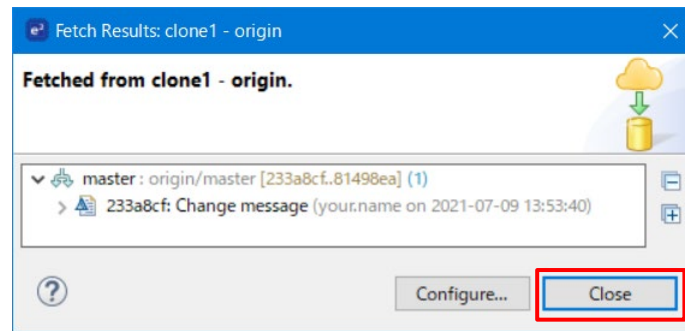


図 29 [Fetch Results] ダイアログ

- 6) 「clone1」の「Remote tracking」ブランチを「clone2」と同じにするためにリベースを実施します。
[Git Repositories] ビューで、「clone1」を選択し、コンテキスト・メニュー [Rebase...] を選択します。
- 7) [Rebase] ダイアログが表示されます。リベースするため、「Remote tracking」からブランチを選択して、[Rebase] ボタンをクリックします。

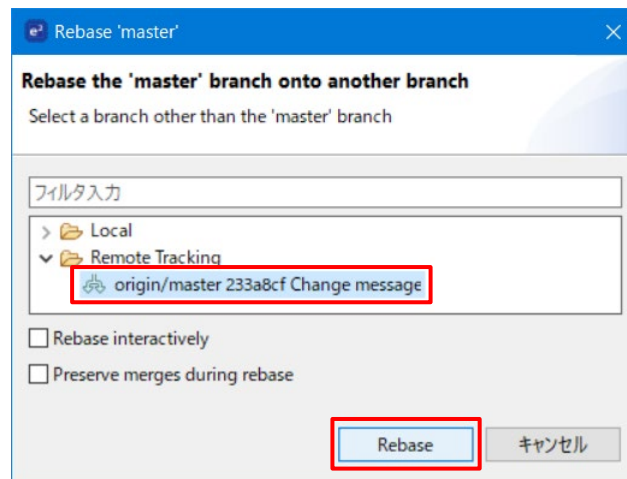


図 30 [Rebase] ダイアログ

- 8) [Rebase Result] ダイアログが表示されます。SampleEGit.c ファイルの競合が表示されます。競合を解決する方法は 4 つあります。
 - 競合を解決してマージする。→ [Start Merge Tool to resolve conflicts]
 - 「clone1」のコミットをスキップする。→ [Skip this commit and continue rebasing the next commits]
 - rebase を中止する。→ [Abort rebase]
 - 何もしない（ワークベンチに戻る）。→ [Do nothing (return to the workbench)]
 ここでは、[Start Merge Tool to resolve conflicts] チェックして、[続行(P)] ボタンをクリックします。

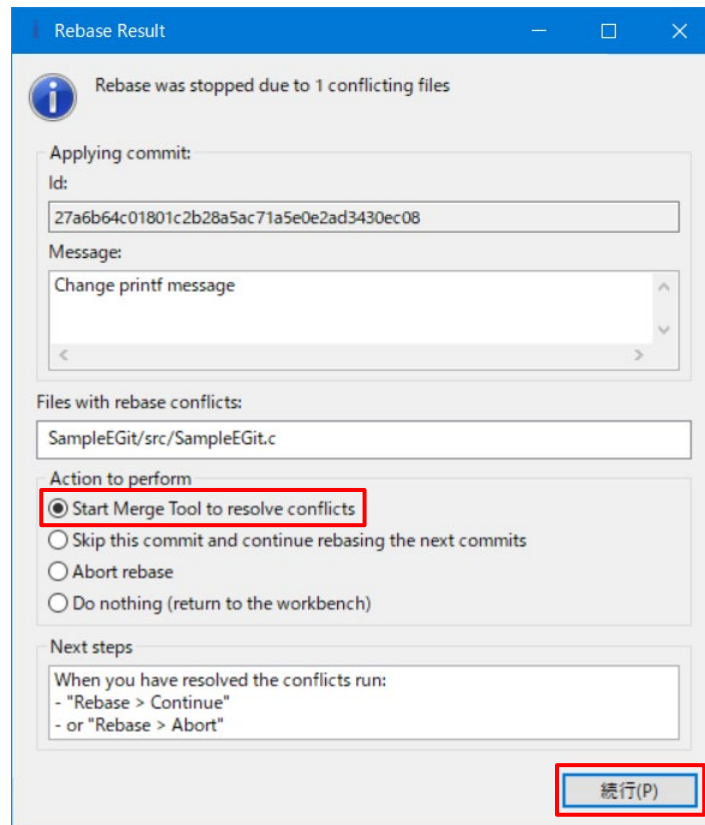


図 31 [Rebase Result] ダイアログ

- 9) [C 比較ビューアー] ビューが表示されます。「clone1」の「Local」ブランチと「Remote tracing」ブランチの違いが表示されます。左側のウィンドウは「Remote Tracking」ブランチの内容、右側のウィンドウは「Local」ブランチの内容が表示されています。右側ウィンドウを参考に、左側ウィンドウの内容を編集して、Ctrl+S キーを押して保存します。

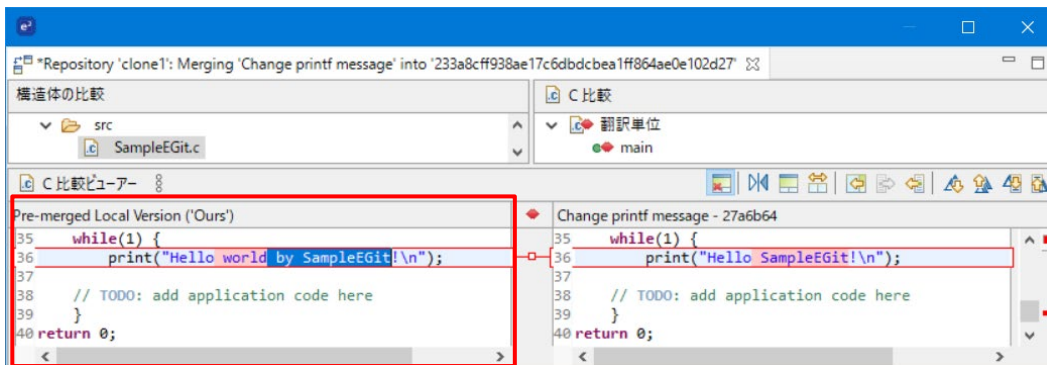


図 32 マージ時の競合内容

- 10) [Git Repositories] ビューで、「clone1」を選択し、[Git Staging] ビューで [Add all files including not selected ones to the index] ボタンをクリックして、[Unstaged Changes] リストボックスの編集ファイルを [Staged Changes] リストボックスに移動します。[Continue] ボタンをクリックして、リベースを完了します。

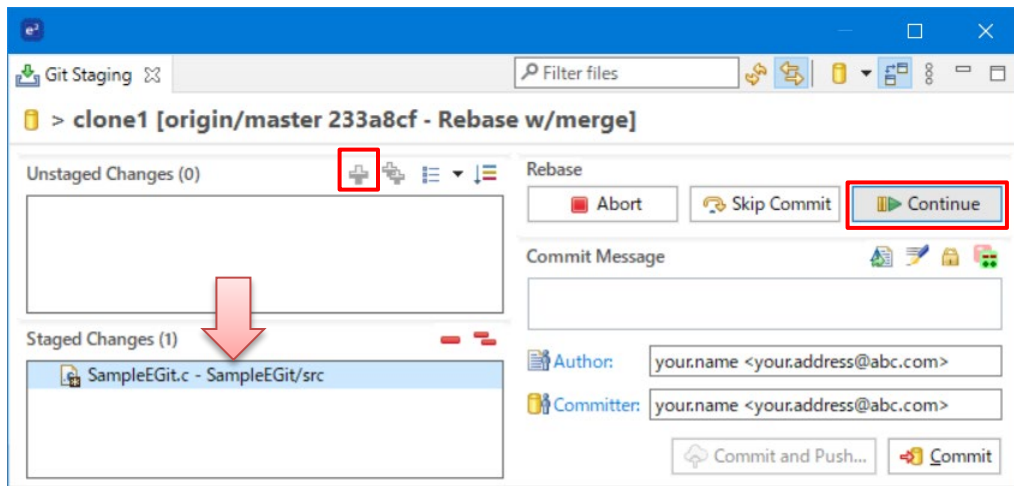


図 33

- 11) [Git Repositories] ビューで、「clone1」を選択し、コンテキスト・メニュー [Push to origin] を選択して、「clone1」の変更をリモート・リポジトリにプッシュします。これにより「clone1」の「Local」ブランチ、および「Remote tracking」ブランチが統合されました。

6. 参考情報

6.1 Web サイト

- EGit | The Eclipse Foundation
<https://www.eclipse.org/egit/>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	Jul.31.21	すべて	・ 英語版 (Rev.1.01) を日本語に翻訳

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 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・システムの設計において、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て生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

- 当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の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。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。
 9. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
 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禁じます。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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が直接的、間接的に支配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。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。