

Coron+, Coron

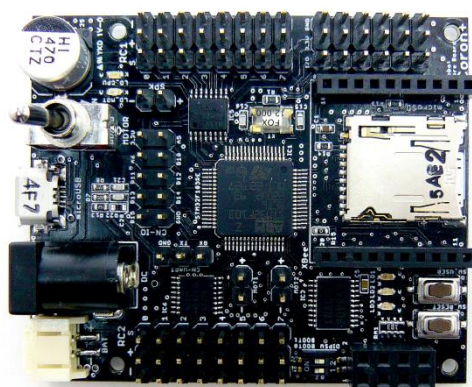
Robot Core Board

IDE3

セットアップマニュアル

v3.0.2 版

はじめにお読みください



TECHNO ROAD Inc.

<http://techno-road.com/>

v3.0.2 2017/12/06

ご使用になる前に

この度は Coron+(以下「Coron」と記載)をお買い上げいただき、誠にありがとうございます。
本製品をお役立て頂くために、このマニュアルを十分お読み頂き、正しくお使い下さい。
今後とも、弊社製品をご愛顧賜りますようよろしくお願いいたします。

*本製品の内容及び仕様は予告なしに変更されることがありますのでご了承ください。

＜取り扱い上の注意＞

- 本製品には、民生用の一般電子部品が使用されております。宇宙、航空、医療、原子力、運輸、交通、各種安全装置など人命、事故に関わる特別な品質、信頼性が要求される用途でのご使用はご遠慮ください。
- 極端な高温下や低温下、または振動の激しい環境での使用はご遠慮ください。
- 水中、高湿度、油の多い環境でのご使用はご遠慮ください。
- 腐食性ガス、可燃性ガス等の環境中でのご使用はご遠慮ください。
- 基板の表面が水に濡れていたり、金属に接触した状態で電源を投入しないでください。
- 定格を超える電源を加えないでください。
- ノイズの多い環境での動作は保証しかねますのでご了承ください。
- 発煙や発火、異常な発熱があった場合には、すぐに電源を切ってください。
- 本書に記載される製品及び技術のうち「外国為替及び外国貿易法」に定める規制貨物等（技術）に該当するものを輸出または国外に持ち出す場合には、同法に基づく輸出許可が必要です。

＜保証について＞

- 本製品の保証は、初期不良交換のみとなります。
万一初期不良品であった場合、商品到着後**30** 日以内に弊社までご連絡ください。不良品と引き換えに交換品をお届けいたします。
- お客様の都合による返品の場合、返送料はお客様負担でお願いいたします。（未使用品に限ります）
- 万が一、本製品を使用して事故または損失が発生した場合、弊社では一切その責任を負いません。
- 本製品の仕様範囲を超える条件において使用された場合については、動作は保証されません。
- 製品を改造した場合、保証は一切適用されません。
- 他社製品との接続互換性および相性問題は保証いたしません。

～目次～

第 1 章	製品について.....	1
第 2 章	開発環境を整える.....	3
第 3 章	プログラムの書き込み.....	4
1.	Eclipse プロジェクトを開く.....	4
	Eclipse の起動.....	4
	サンプルプロジェクトのインポート.....	6
	プロジェクトを開く.....	10
2.	プログラムをビルドする.....	14
3.	Coron にプログラムを書き込む（USB 編）.....	18
	Coron を DFU モードにする.....	18
	DFU ドライバーのインストール.....	20
	プログラムの書き込み.....	28
	USB CDC ドライバーのインストール.....	31
第 4 章	付属サンプルプログラムについて.....	36
	Eclipse に新しくプロジェクトを追加する.....	37

第1章 製品について

※Coronに関する最新情報は → <http://techno-road.com/board.html>

内容物一覧

- ・ Coron+（または Coron）本体(以下まとめて「Coron」と記載)
- ・ USB ケーブル(A-microB タイプ)(Coron の場合は A-miniB)
- ・ カード（開発環境、マニュアル等の zip ファイルダウンロード URL とパスワードが記載されています）

ダウンロード URL の各項目の内容

<Coron+、Coron 共通>

- ・ **【IDE3_manual】**
 - IDE3 セットアップマニュアル.pdf **本書**
 - Coron・Coron_plus 共通ソフトウェアライブラリ.pdf
 - JTAG マニュアル.pdf
 - 従来の Coron(無印)と Coron プラスの対応表.pdf
- ・ **【Coron IDE3】**
 - Coron IDE3.zip（開発環境）

<Coron+>

- ・ **【coronplus_board】**
 - Coron_plus ボードマニュアル.pdf
 - 回路図.pdf
- ↳ **【datasheet】**
 - CPU ハードウェアマニュアル(英語版).pdf
 - CPU ハードウェアマニュアル(日本語版).pdf
 - CPU ソフトウェアマニュアル(英語版).pdf
 - CPU ソフトウェアマニュアル(日本語版).pdf

<旧 Coron（Coron(無印)）>

- ・ **【coron_board】**
 - Coron ボードマニュアル.pdf
 - 回路図.pdf
- ↳ **【datasheet】**
 - CPU ハードウェアマニュアル(英語版).pdf
 - CPU ハードウェアマニュアル(日本語版).pdf
 - CPU ソフトウェアマニュアル(英語版).pdf
 - CPU ソフトウェアマニュアル(日本語版).pdf

推奨 PC 動作環境

OS : Windows 7 / 8.1 / 10

CPU : Intel Pentium プロセッサ 1GHz 以上

HDD 空き容量 : 800MB 以上

メモリ : 512MB 以上

ソフトウェアの構成

エディター : Eclipse

コンパイラ : GCC:MinGW & ARM ツールチェーン:GNU ARM

書き込みソフト : USB…CoronDFU JTAG…openOCD

第2章 開発環境を整える

Coron+（または Coron 以下まとめて「Coron」と記載）のプログラム開発環境を整える方法について解説します。

既に、Coron 付属のカードに記載されている URL より、「IDE3_manual.zip」、「CoronIDE3.zip」、「coronplus_board.zip（または coron_board.zip）」の3つの zip ファイルをダウンロードしていると思います。

そのうちの CoronIDE3.zip を使用します。

CoronIDE3.zip を、C ドライブの直下 (C:¥) や、D ドライブの直下(D:¥)に CoronIDE3 フォルダごと解凍して下さい。（特定の日本語、特殊文字等で動かないことがあります。C や D ドライブ直下以外での動作保障は致しませんのでご注意ください。）

これで Coron の開発環境は整いました。

※CoronIDE3 からは、以前のような開発環境のインストールは必要なくなりました。ファイルを解凍するだけで準備は完了します。

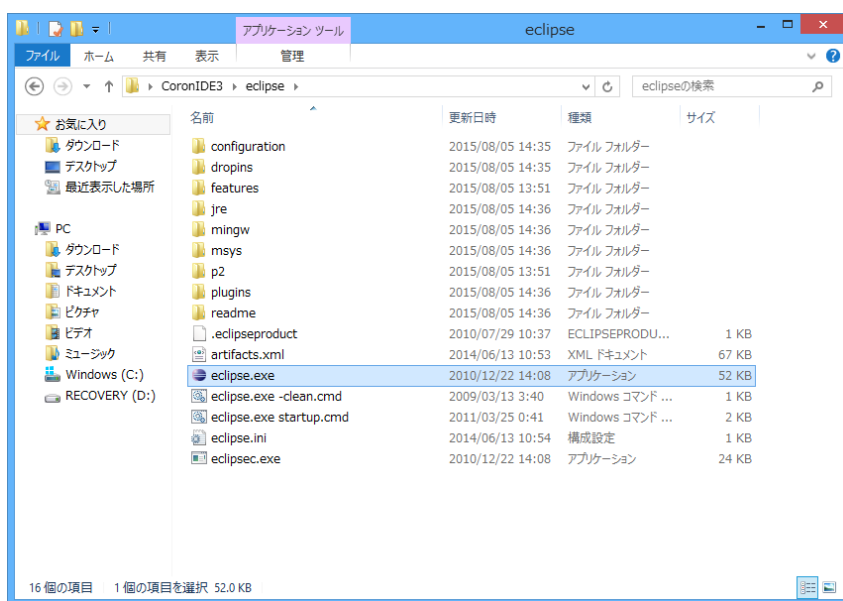
第3章 プログラムの書き込み

付属のサンプルプログラムを例にして、実際に開発環境を開いてプログラムを Coron に転送・起動するところまでを紹介します。

1. Eclipse プロジェクトを開く

Eclipse の起動

まず、先ほど解凍した【Coron IDE3】フォルダ内にある【eclipse】フォルダより、【eclipse.exe】を起動して下さい。



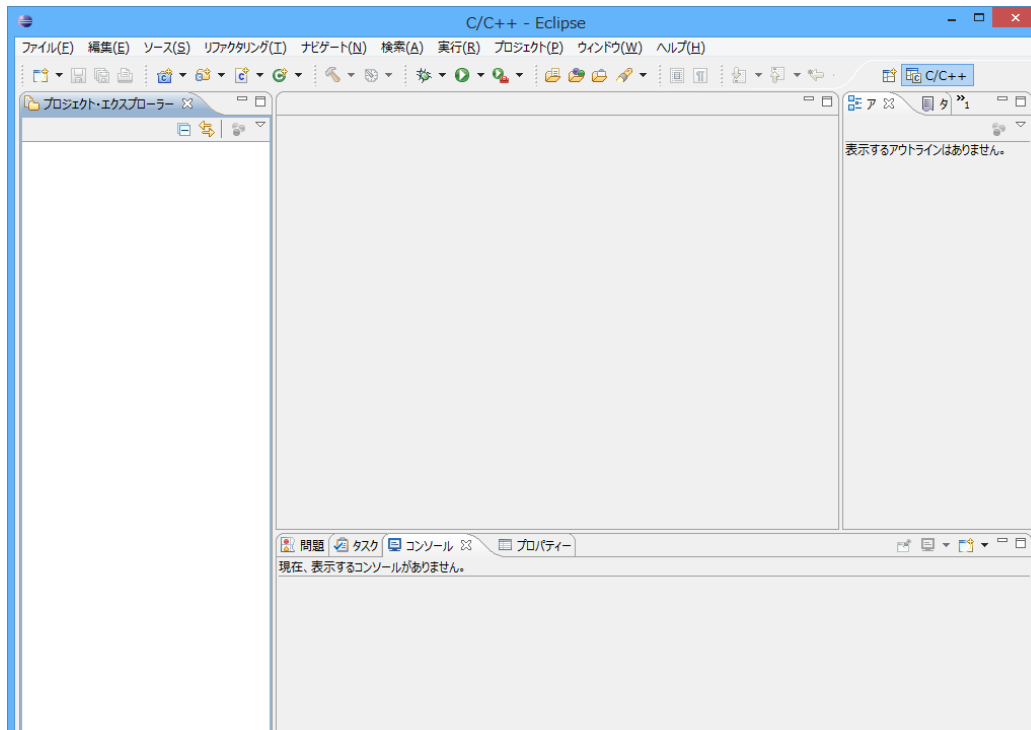
すると下図のような画面が表示され Eclipse が起動します。



起動すると、「ようこそ」というタブが表示されるので、これは×を押して閉じて下さい。

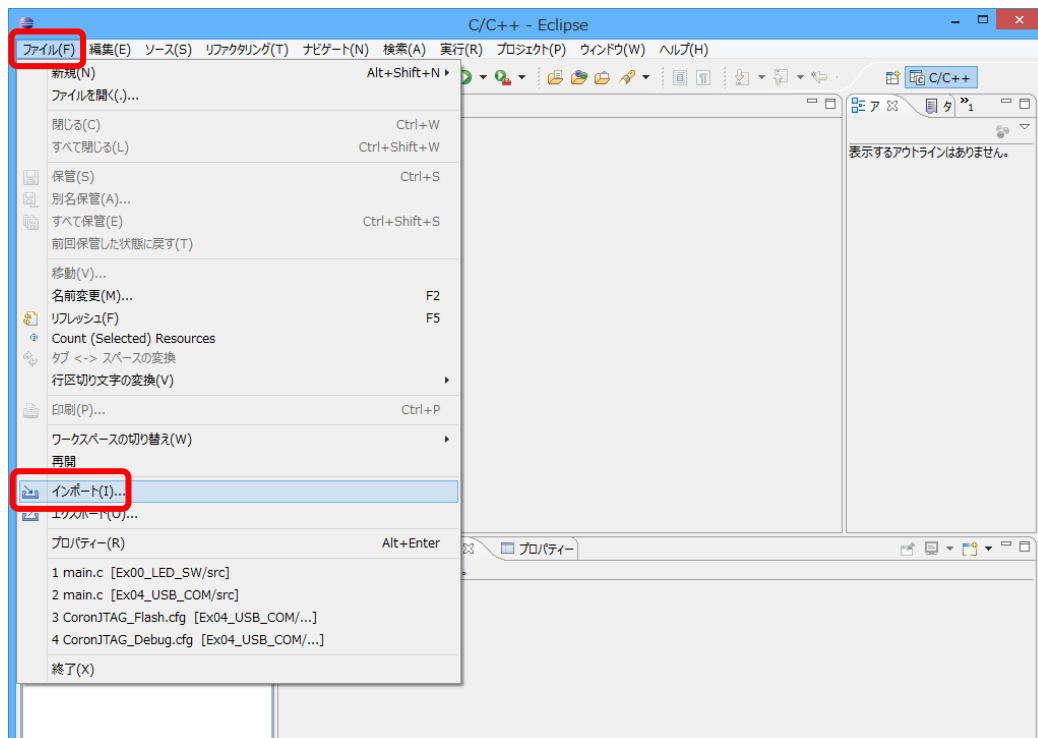


しばらく待つと下図のようなメイン画面が表示されます。プログラミング作業は全てこのEclipse 上で行われます。

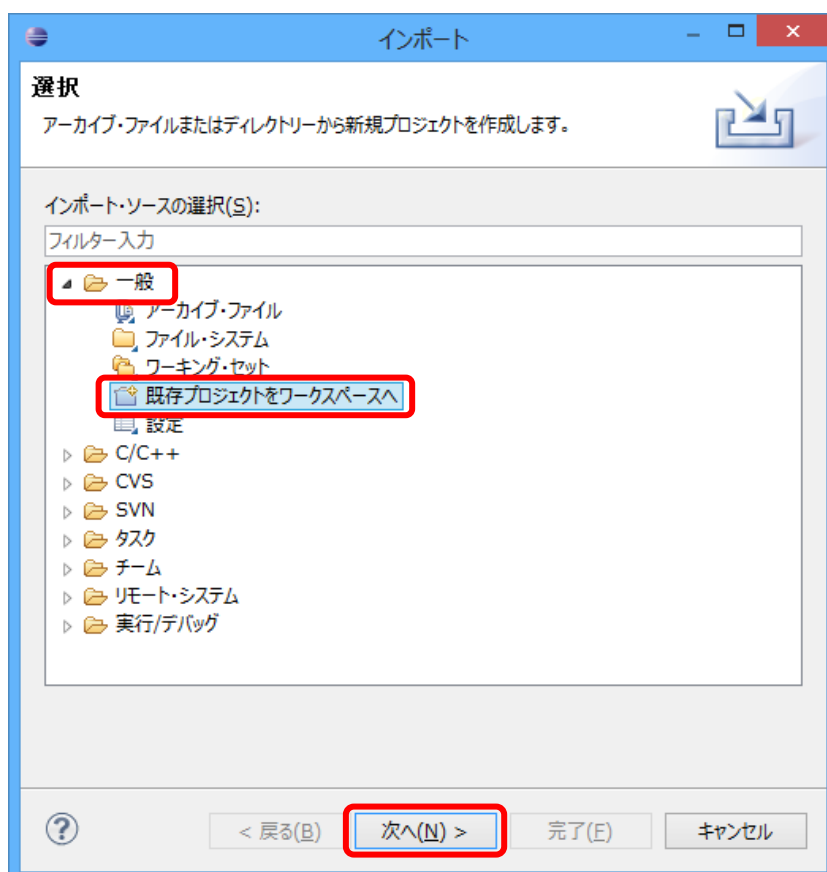


サンプルプロジェクトのインポート

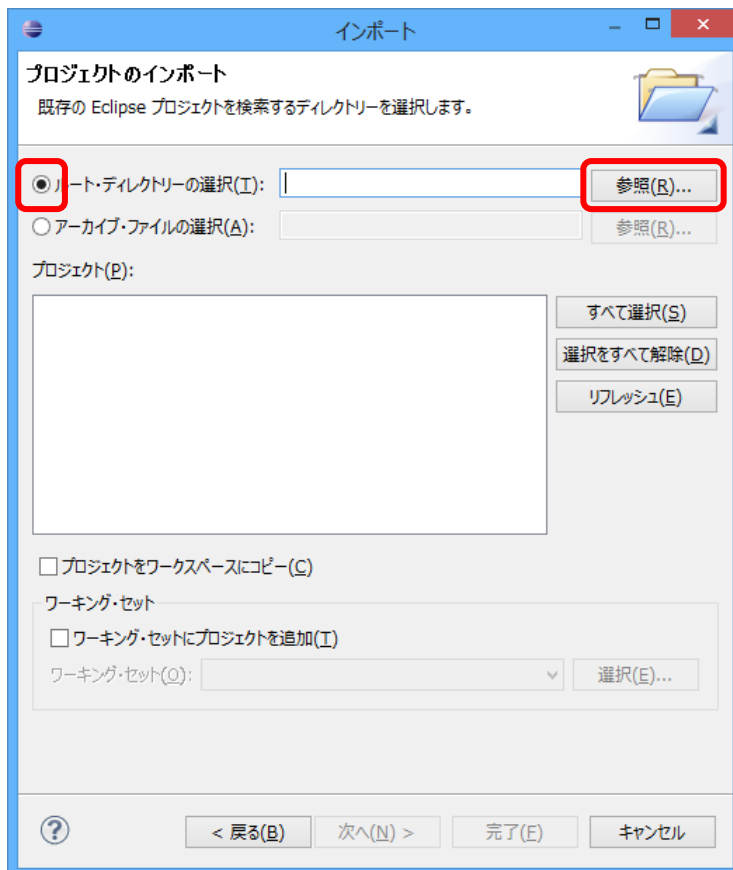
Eclipse 上で、【ファイル】→【インポート】を選択して下さい。



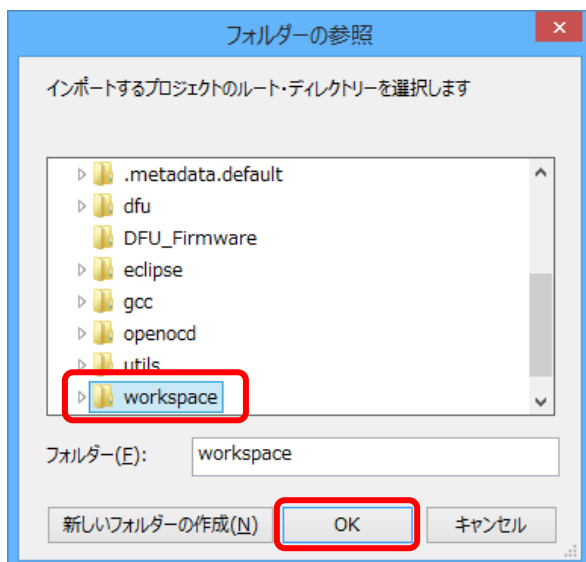
【一般】→【既存プロジェクトをワークスペースへ】を選択して【次へ】をクリックしてください。



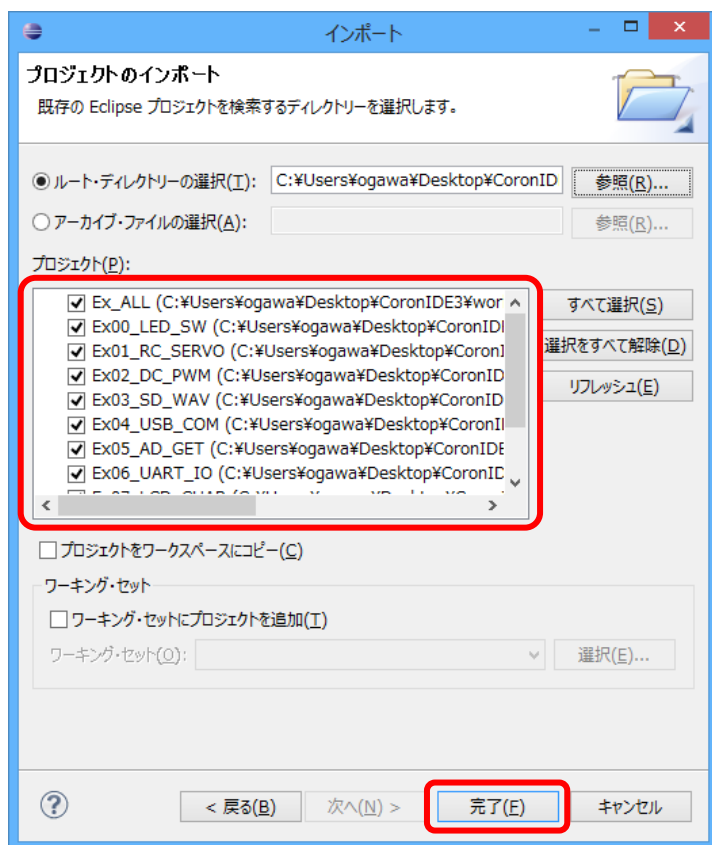
【ルート・ディレクトリーの選択】にチェックをし、【参照】を選択してください。



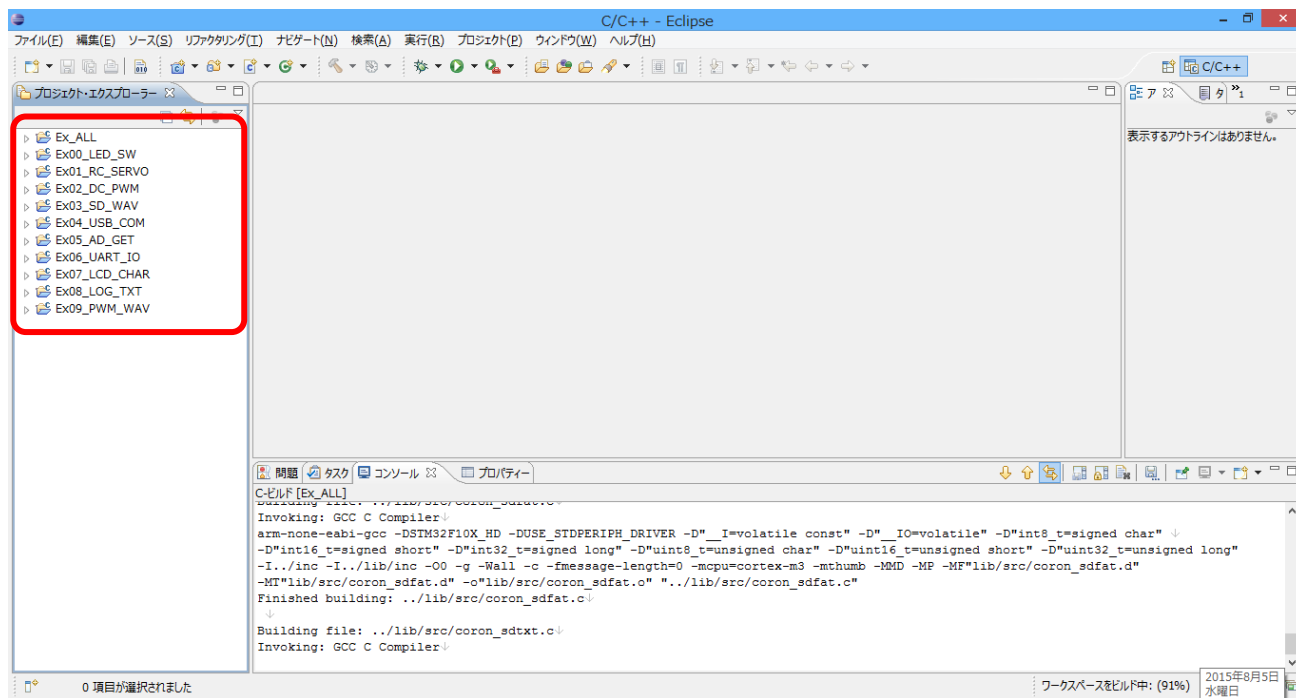
【workspace】を選択し、【OK】をクリックして下さい。



「プロジェクト」覧に「Ex_ALL」、「Ex00_LED_SW」から「Ex09_PWM_WAV」までが表示されます。それぞれにチェックが入っていることを確認し、【完了】をクリックして下さい。



プロジェクトがインポートされます。

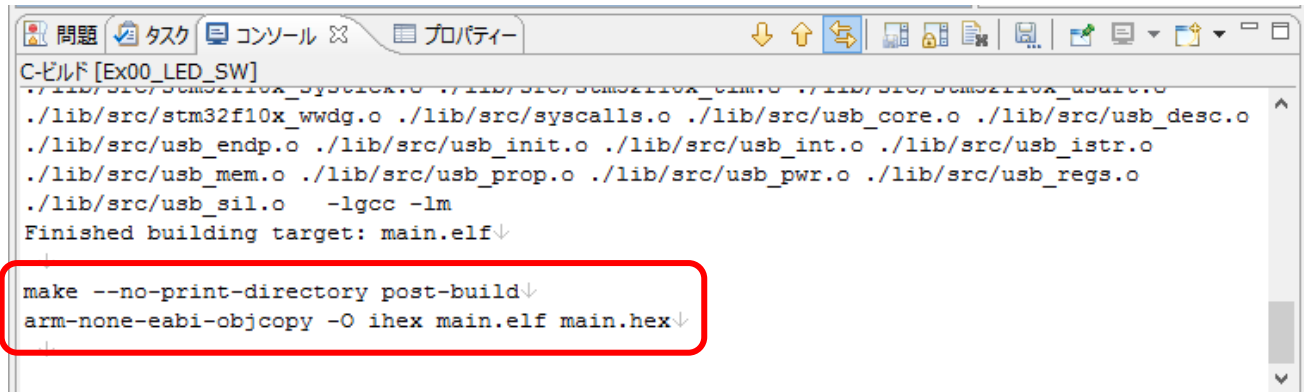


プロジェクトをインポートすると、自動的にビルドが実行されるので、Eclipse の「コンソール」タブに「make --no-print...

arm-none-eabi-objcopy...

が表示され停止するまで待ってください。

これでサンプルプロジェクトを使用できるようになりました。



```
C-ビルド [Ex00_LED_SW]
./lib/src/stm32f10x_syscfg.o ./lib/src/stm32f10x_tim.o ./lib/src/stm32f10x_usart.o
./lib/src/stm32f10x_wwdg.o ./lib/src/syscalls.o ./lib/src/usb_core.o ./lib/src/usb_desc.o
./lib/src/usb_endp.o ./lib/src/usb_init.o ./lib/src/usb_int.o ./lib/src/usb_istr.o
./lib/src/usb_mem.o ./lib/src/usb_prop.o ./lib/src/usb_pwr.o ./lib/src/usb_regs.o
./lib/src/usb_sil.o -lgcc -lm
Finished building target: main.elf↓
make --no-print-directory post-build↓
arm-none-eabi-objcopy -O ihex main.elf main.hex↓
```

プロジェクトを開く

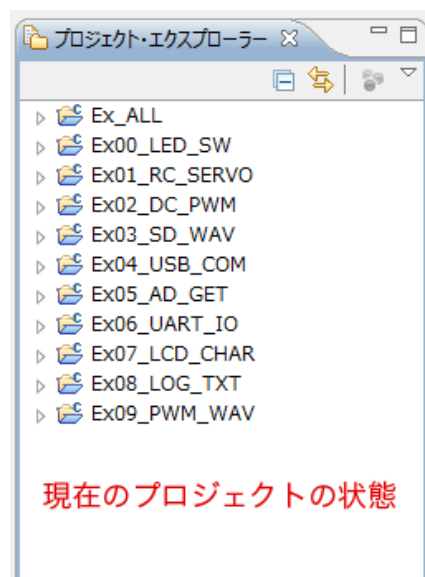
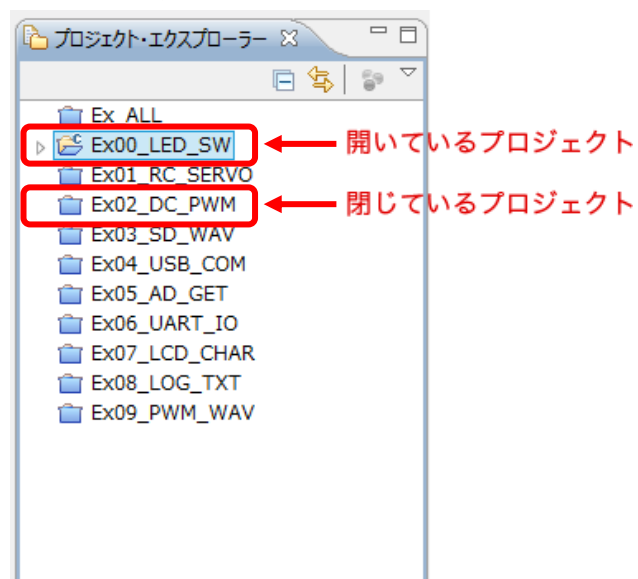
次は、インポートしたサンプルを使って、**Coron** にプログラムの書き込みをします。

Eclipse の基本的な使い方を確認しながら説明していきます。

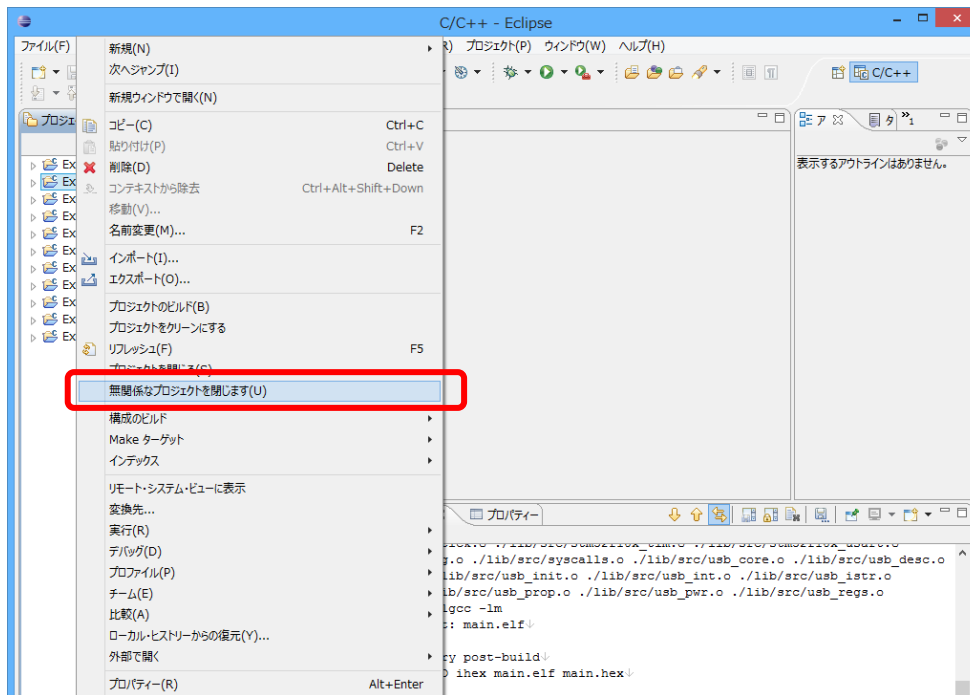
Eclipse の画面左にあるのがプロジェクト・エクスプローラーです。

プロジェクトの開閉や追加・削除、またプロジェクト内の各プログラムファイルをここで一括管理します。

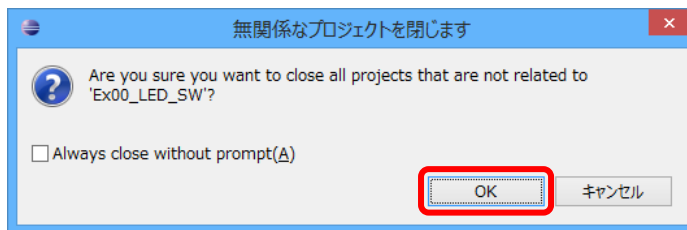
プロジェクトには開いた状態のプロジェクトと、閉じているプロジェクトがあり、プロジェクト名の隣にあるフォルダのアイコンで開閉状態を判断できます。



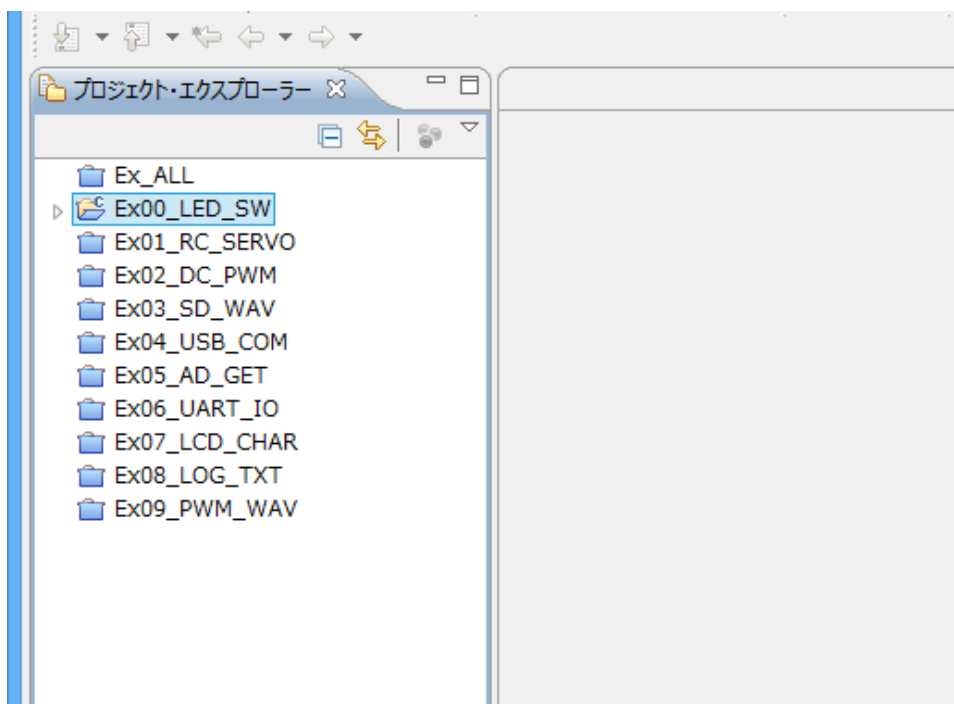
今はサンプルプロジェクトが全て開いている状態なので、使用するプロジェクト（ここでは、「Ex00_LED_SW」というプロジェクトを選択）を左クリックで選択し、右クリック→【無関係なプロジェクトを閉じます】を選択してください。



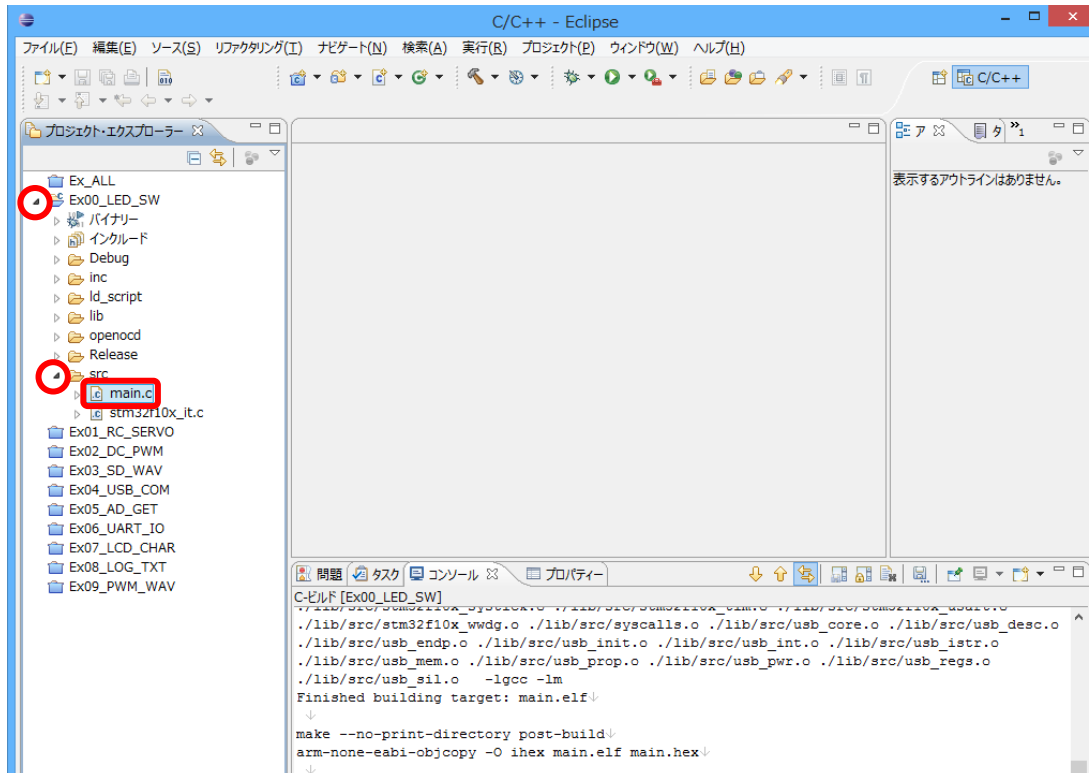
「無関係なプロジェクトを閉じます」というダイアログが表示されたら、【OK】を選択し、使用するプロジェクト以外のプロジェクトを全て閉じます。



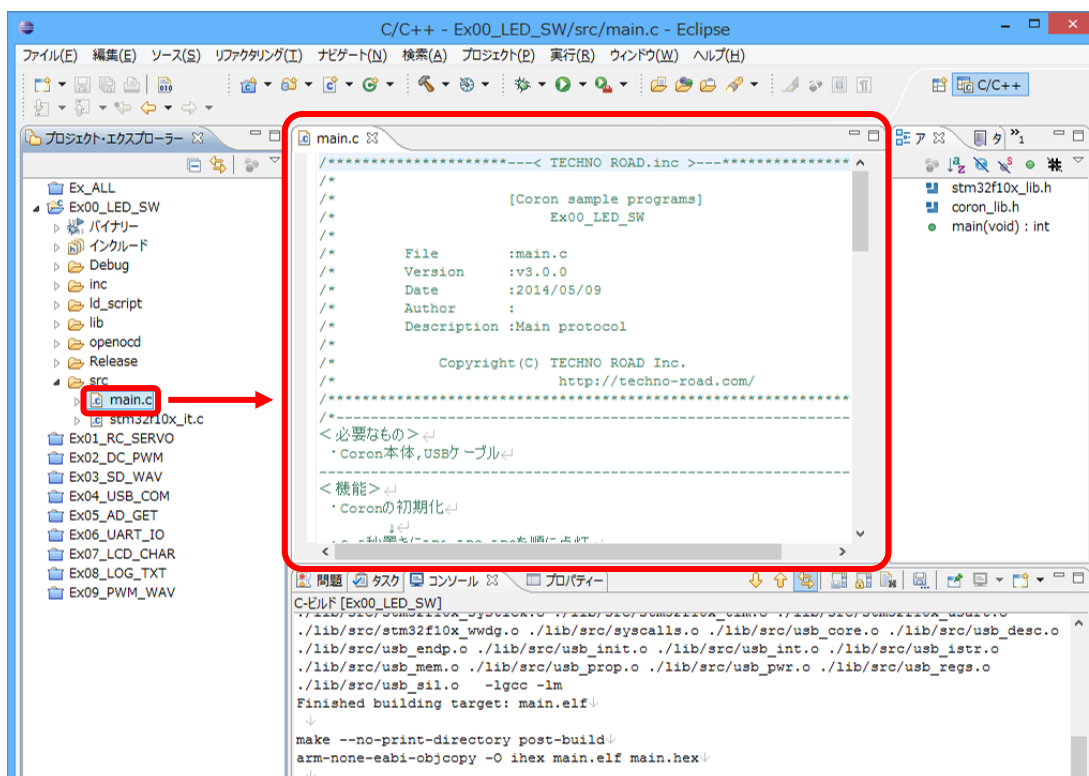
「Ex00_LED_SW」のみ開いている状態です。編集集中のプロジェクトがわかりやすくなりました。



「Ex00_LED_SW」フォルダの左にある三角矢印をクリックし、【src】→【main.c】を選択しダブルクリックします。「main.c」に、プログラムの中身が書かれています。



ファイルの名前をダブルクリックすることで画面右側のエディター部分に内容が表示されます。





注意

旧 Coron（USB が miniB のもの）をご使用の場合、inc¥coron_conf.h の

```
#define _CORON_PLUS //Coron+を使用
```

を下記のようにコメントアウトして下さい。

```
//#define _CORON_PLUS //Coron+を使用
```

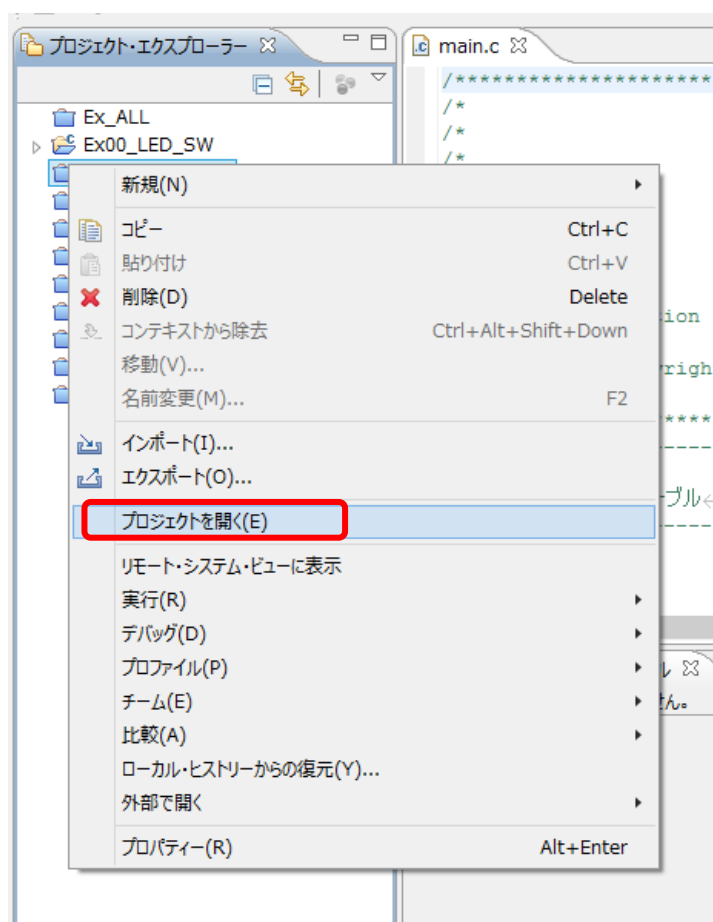
Coron+をお使いの方はコメントアウトせずそのままご使用下さい。



ワンポイント

プロジェクトを開くときも同じように、開きたいプロジェクトを選択して右クリック→【プロジェクトを開く】を選択すると、プロジェクトが開きます。（※閉じる場合も同様に、プロジェクトを選択して右クリック→【プロジェクトを閉じる】を選択して下さい。）

※プログラム編集時に開いているプロジェクトは、一つだけにすることを推奨します。
（編集中のプロジェクトをわかりやすくするため）



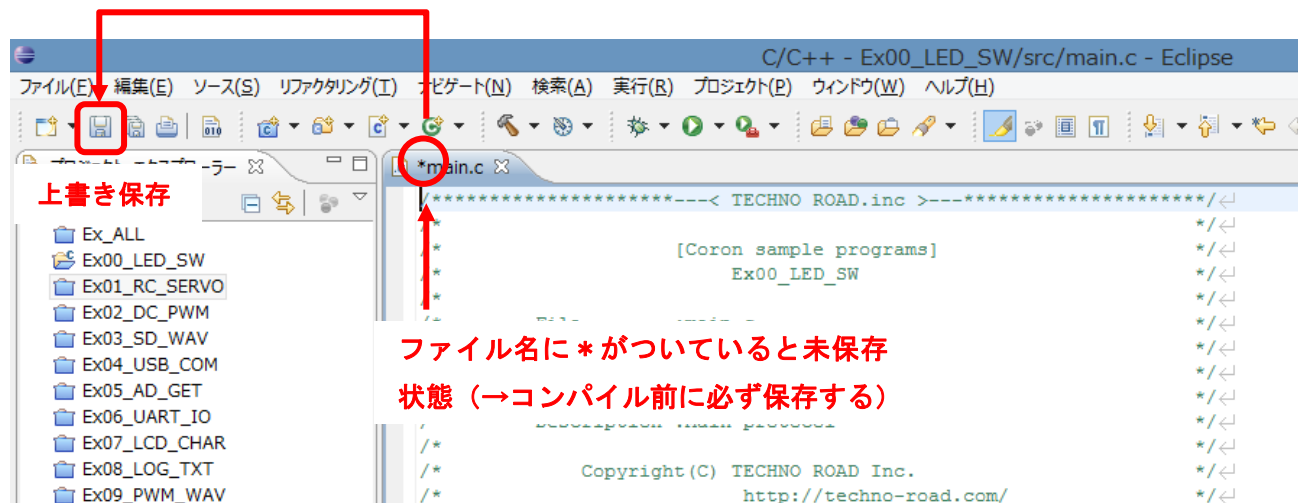
2. プログラムをビルドする

プログラムを Coron に書き込む際にはビルド（コンパイル）する必要があります。

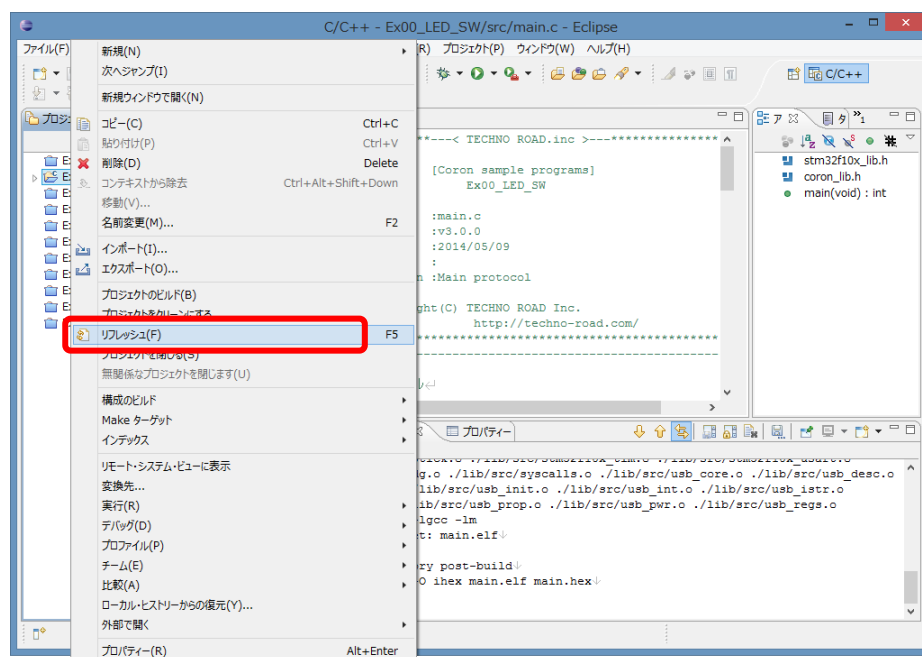
まず、下図のようにファイル名に*が付いていないかをチェックします。

付いている場合にはそのファイルは編集後まだ保存されていないということなので、保存したいフォルダ（ここでは「Ex_00_LED_SW」）を選択した状態で、必ず上書き保存をして下さい。

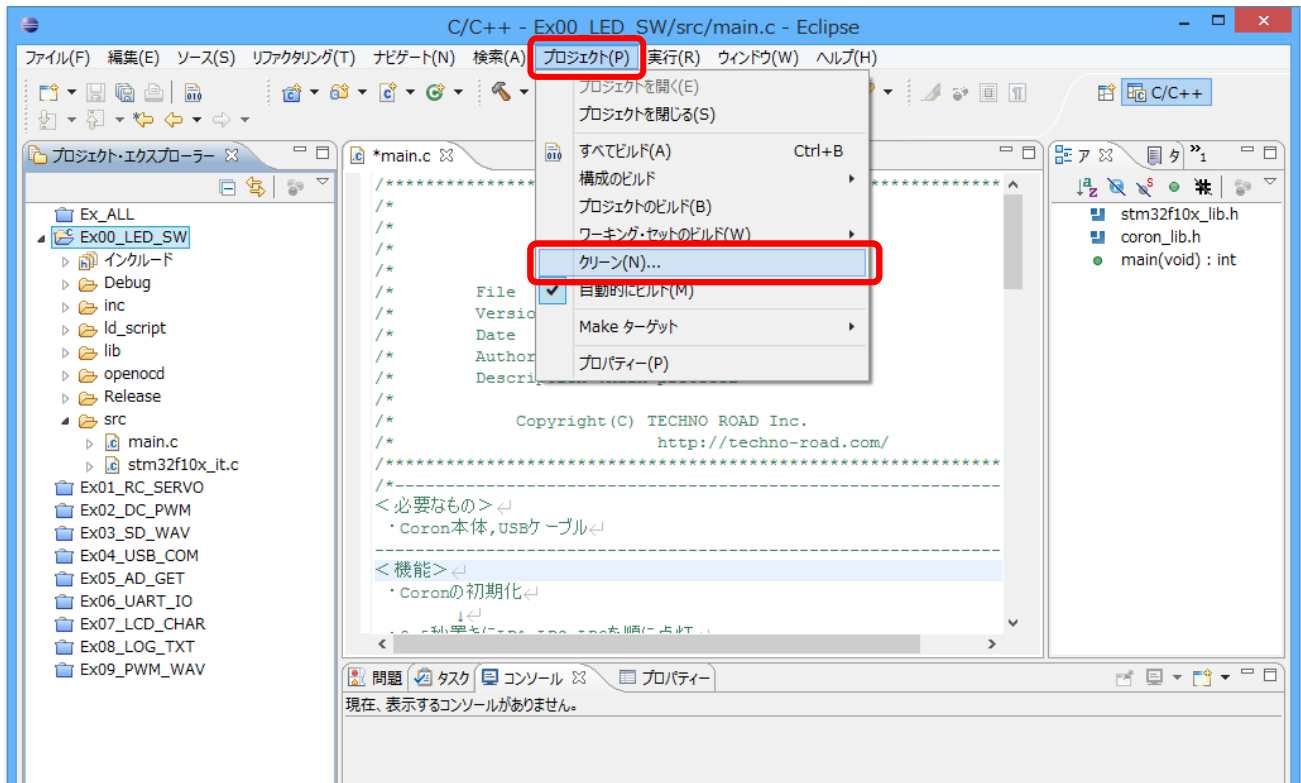
メニューバーに上書き保存アイコンがあるので、クリックして保存して下さい。



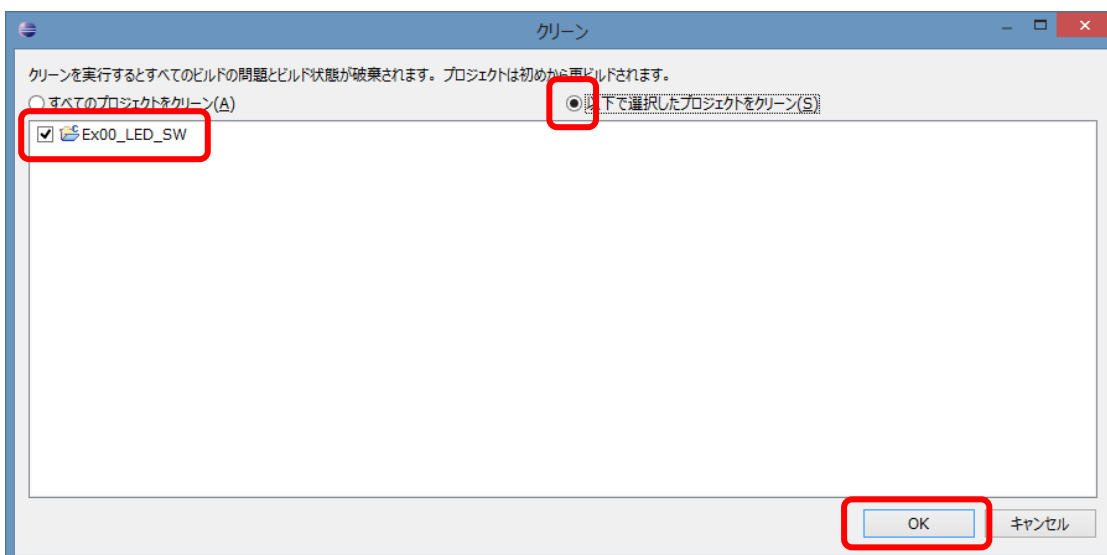
次に【Ex00_LED_SW】を右クリック→【リフレッシュ】でプロジェクトをリフレッシュして下さい。



次は【プロジェクト】→【クリーン】を選択して下さい。



「クリーン」のダイアログが表示されたら、【以下で選択したプロジェクトをクリーン】にチェックをし、【Ex00_LED_SW】にチェックをして、【OK】を選択してプロジェクトをクリーンして下さい。

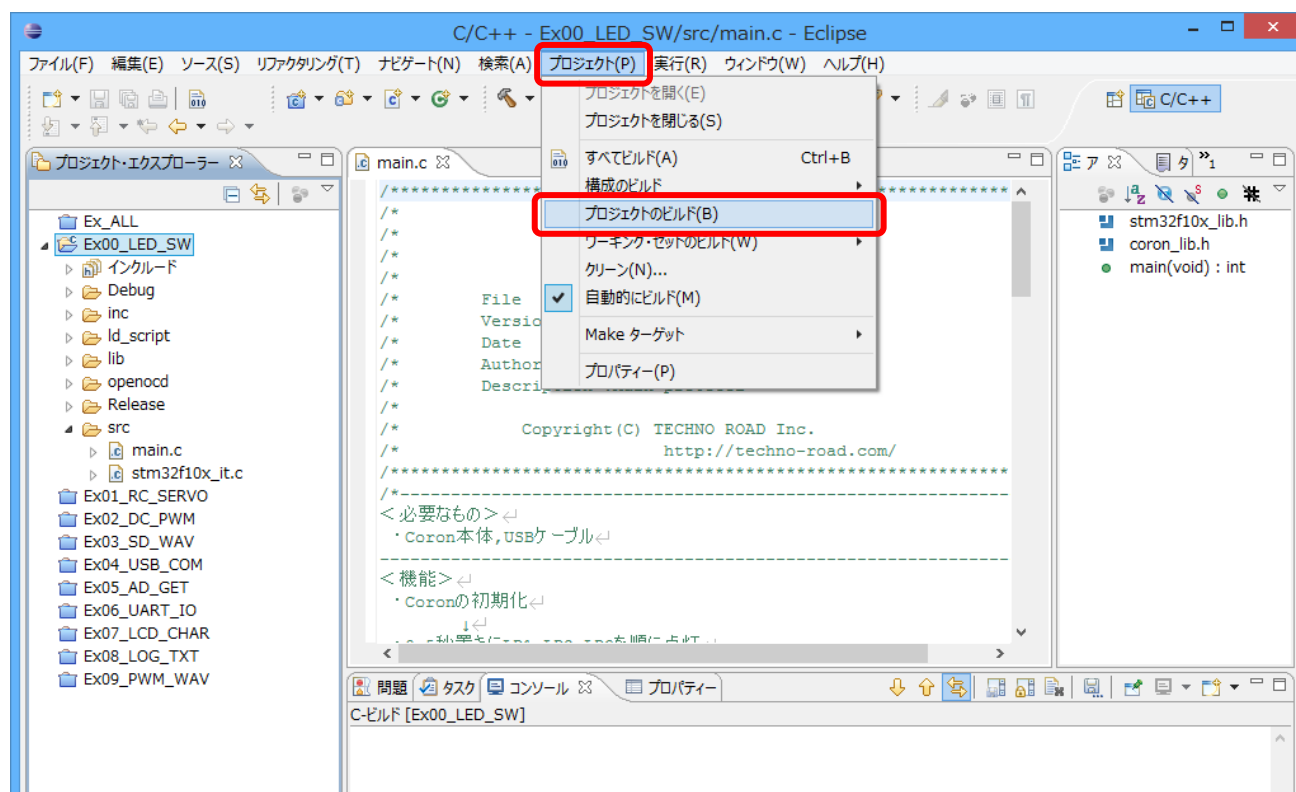


※リフレッシュとクリーンは基本的に下記の赤枠内の条件に当てはまる場合のみ実行すれば十分です。その後の作業ではリフレッシュ→クリーンを省略しても構いません。

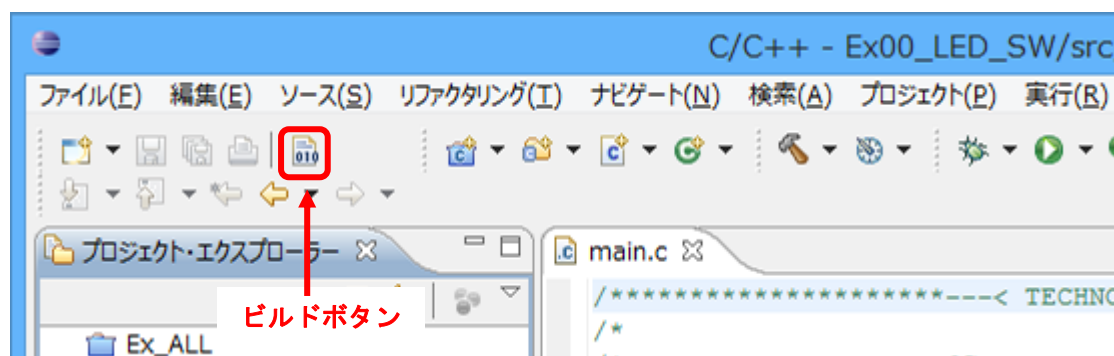
※下記の場合は今の手順でリフレッシュ→クリーンをするようにして下さい。

- プロジェクトを大幅に変更してから最初にビルドする前
- プロジェクトをインポートし、初めて開いた時
- ビルド（コンパイラ）の調子がおかしいとき
- プロジェクト内にファイルを追加もしくは削除した場合

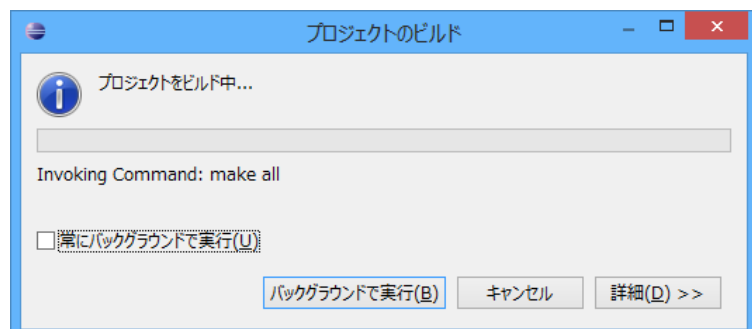
次に、【Ex00_LED_SW】を選択した状態で、【プロジェクト】→【プロジェクトのビルド】を選択して、プロジェクトをビルドして下さい。



※メニューバーにある、ビルドボタンを押してビルドしても同様です。

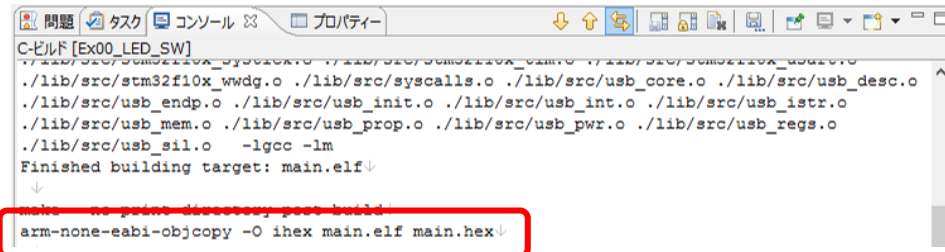


ビルドが開始されると「プロジェクトのビルド」というポップアップが表示され、進行状況がわかります。



初回起動時は全てのプログラムソースが自動的にビルドされますが、次回以降はユーザによって編集が行われたプログラムソースだけをビルドして下さい。

ポップアップが閉じ、コンソール画面に「**arm-none-eabi-objcopy -O ihex main.elf main.hex**」と表示されたら無事ビルド終了です。



```

C-ビルド [Ex00_LED_SW]
../lib/src/stm32f10x_wwdg.o ../lib/src/syscalls.o ../lib/src/usb_core.o ../lib/src/usb_desc.o
../lib/src/usb_endp.o ../lib/src/usb_init.o ../lib/src/usb_int.o ../lib/src/usb_istr.o
../lib/src/usb_mem.o ../lib/src/usb_prop.o ../lib/src/usb_pwr.o ../lib/src/usb_regs.o
../lib/src/usb_sil.o -lgcc -lm
Finished building target: main.elf
make: no print directory post build
arm-none-eabi-objcopy -O ihex main.elf main.hex

```

ビルドが完了するとプロジェクトの **Debug** フォルダの中に **main.hex** が生成されます。

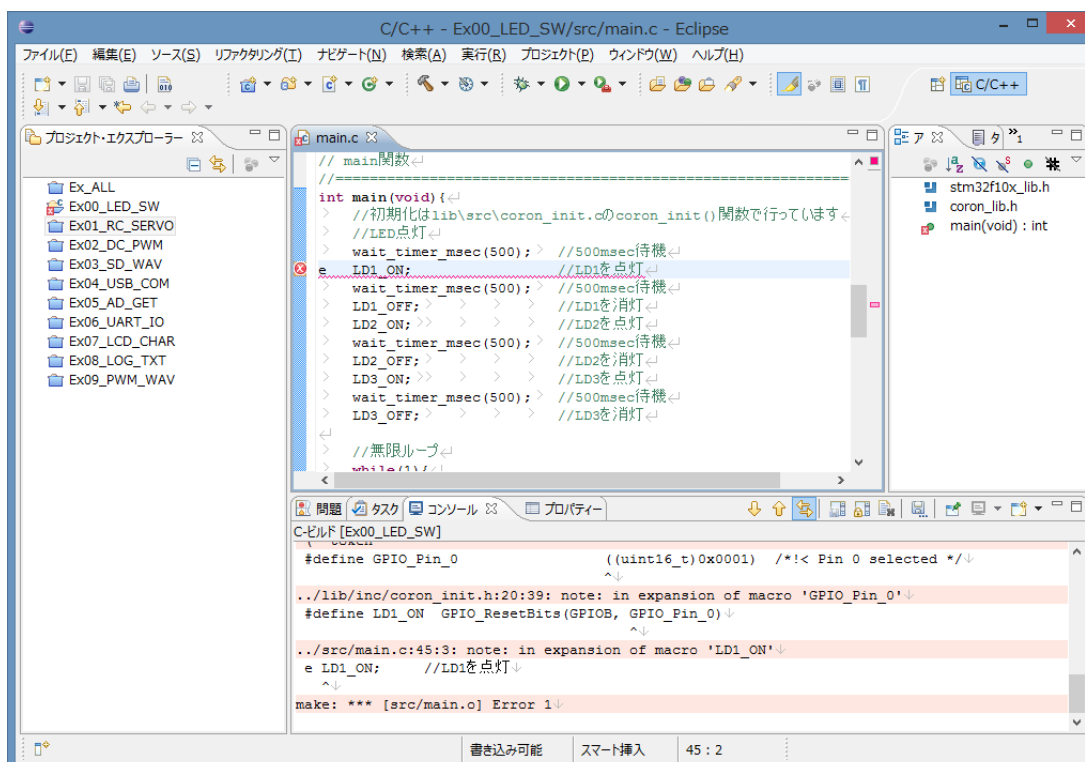
ここでコンパイルされたプログラムを、USB (DFU) もしくは JTAG どちらかの方式で Coron に書きこむことになります。

次節で USB (DFU) での書き込み方法について示します。(本誌では JTAG での書き込み方法は記載しておりません。JTAG で書きこむ場合は別紙の「JTAG マニュアル.pdf」を参照して下さい。



エラーが出た時

リフレッシュ、クリーン、ビルドを行う最中に、エラーが出た場合は、コンソールの指示通りにプログラムを修正して、ビルドをもう一度やり直してください。(ビルドが失敗したファイルは赤字のチェックマークが付きます)



3. Coron にプログラムを書き込む (USB 編)

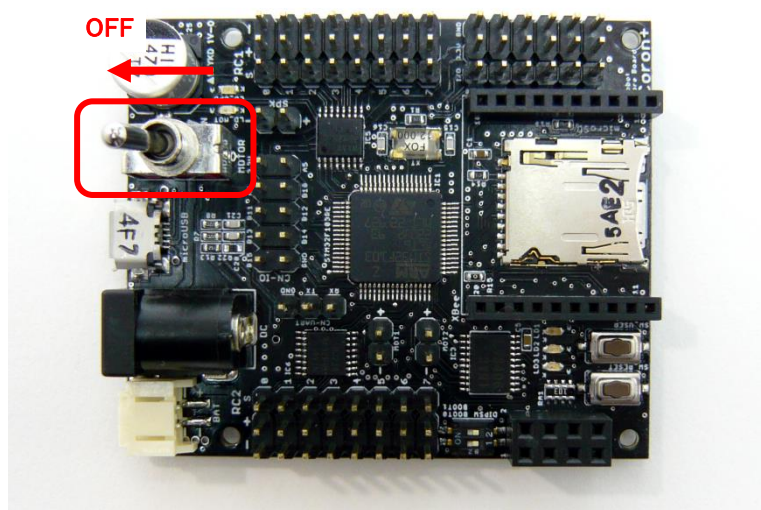
先ほどビルドして生成された Hex ファイル「main.hex」を USB で Coron に転送する方法を説明します。(ビルドの方法は 14 ページのプログラムをビルドするを参照して下さい。)

Coron を DFU モードにする

Coron には出荷時に DFU(Device Firmware Upgrade)というソフトが書き込まれています。このソフトは、Coron が「DFU モード」になっている間、USB 接続でプログラムを書き込めるといふソフトです。

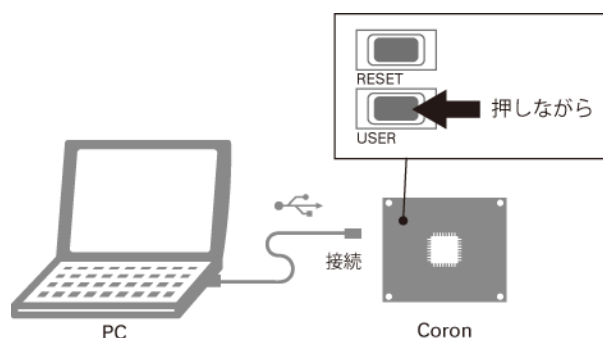
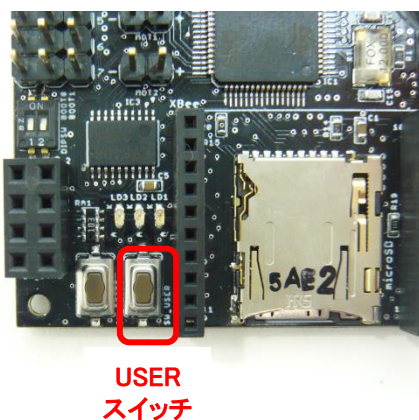
では、Coron を DFU モードにする方法を説明します。

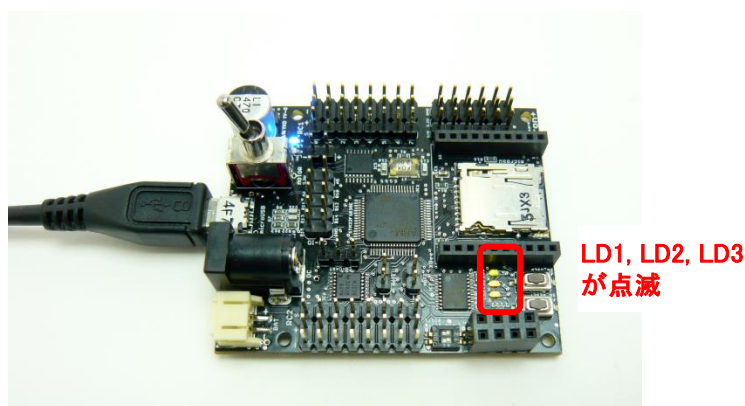
MOTOR 電源スイッチを OFF にします。



次に、下記写真のように、Coron の SW USER を押しながら電源を ON (USB ケーブルで PC と接続) すると、LD1・LD2・LD3 が同時に点滅を始めます。

この点滅が DFU モードに入った合図です。

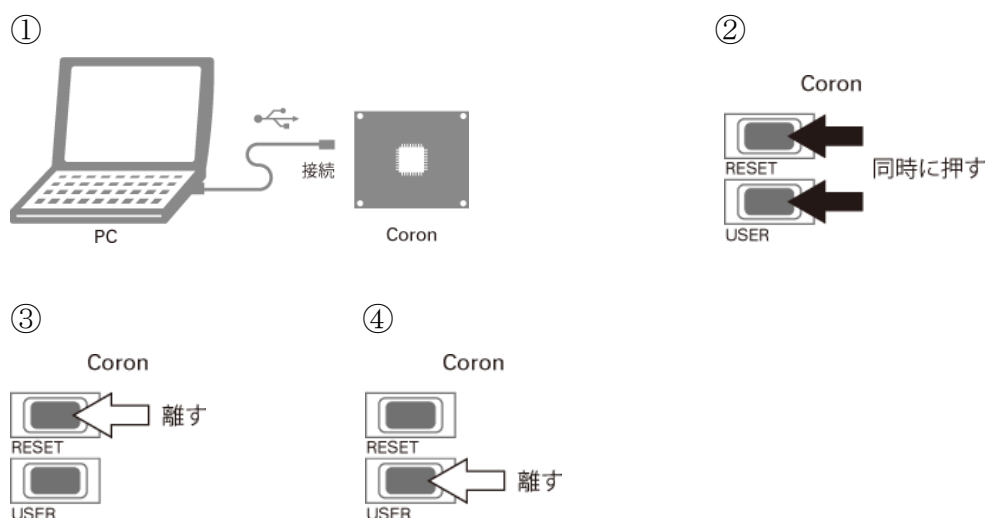




点滅が始まらない場合は、DFU モードに入っていないので、もう一度 SW_USER を押しながらケーブルで接続してください。

尚、CoronとPCを（何も押さずに）USBケーブルで接続した後、CoronのSW_RESETとSW_USERを同時に押し、SW_RESET→SW_USERの順に離すことでもDFUモードに入ることが可能です。（下図参照）

※このDFUへの移行方法は、旧Coron（USBがminiBのもの）及びCoron+のシリアルNo「F0902」では使用できません。

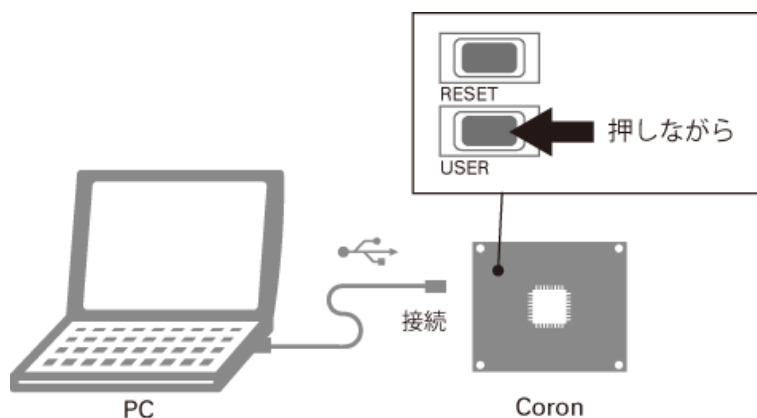


DFUモードに入ったCoron をPCと接続する際、**初回接続時のみ「DFUドライバー」をインストールする必要があります。**次項のDFUドライバーのインストールより手順に従って、ドライバーをインストールして下さい。

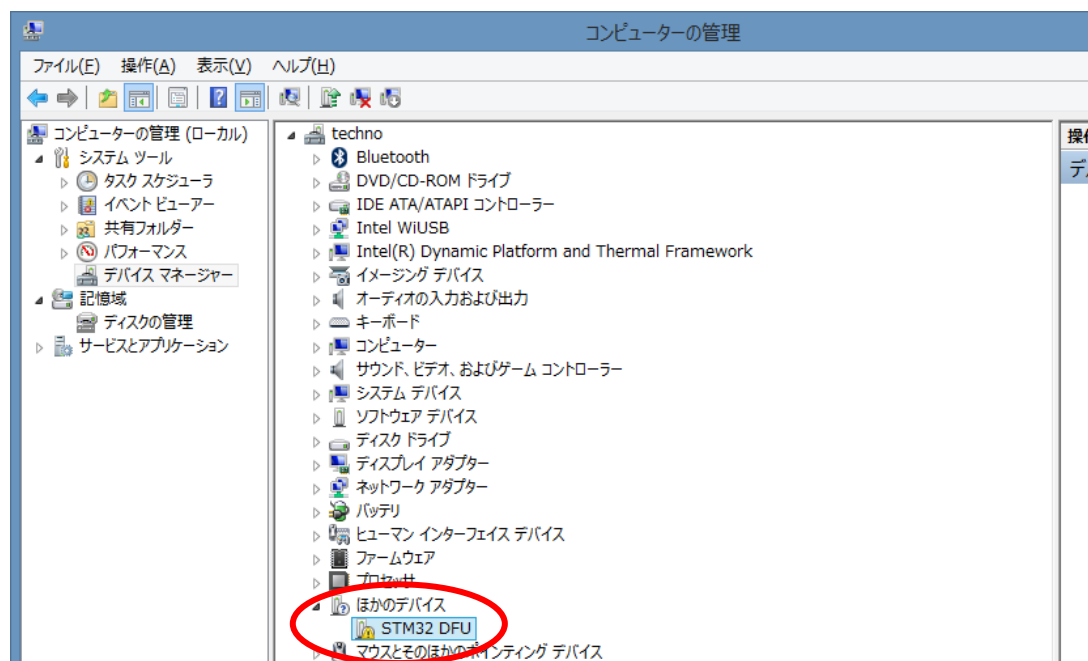
なお、CoronとPCをUSBで接続しCOM Portとして通信するプログラムを作りたい場合（EX04_USB_COM等）は、31ページの**USB CDCドライバーのインストール**も必要となりますので、お忘れのないようご注意ください。

DFU ドライバーのインストール

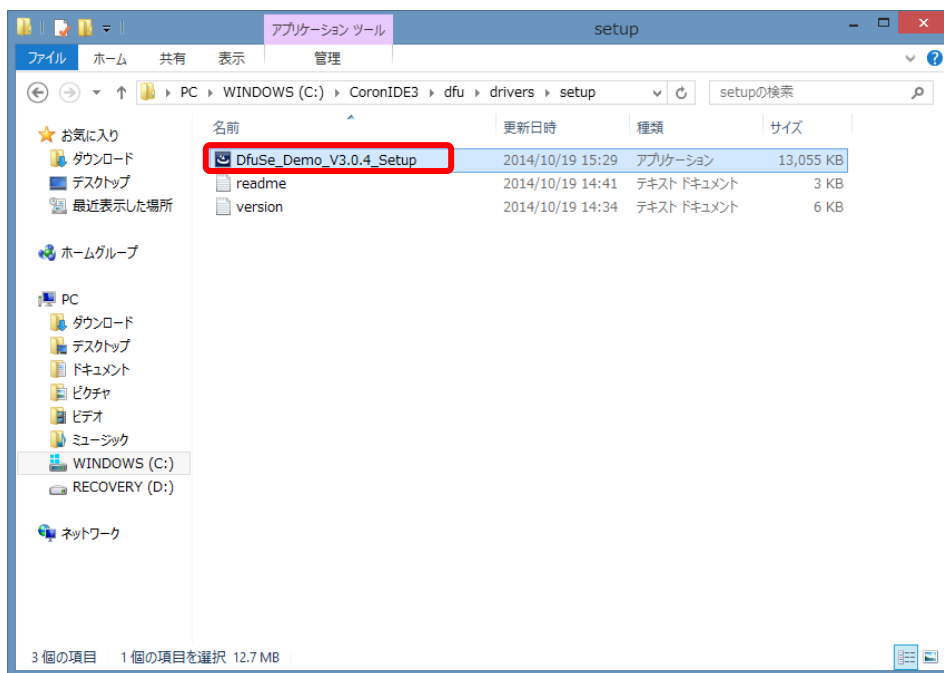
1. Coron を DFU モードで PC に USB 接続してください。(SW_USER を押しながら PC に USB 接続すると DFU モードになります。)



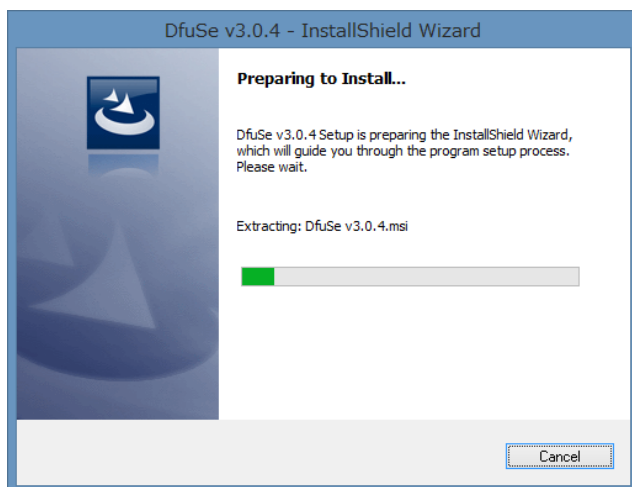
2. 【PC】→【管理】で「コンピューターの管理」を開き、【デバイスマネージャー】→【ほかのデバイス】を選択します。【STM32 DFU】と表示されているか確認して下さい。



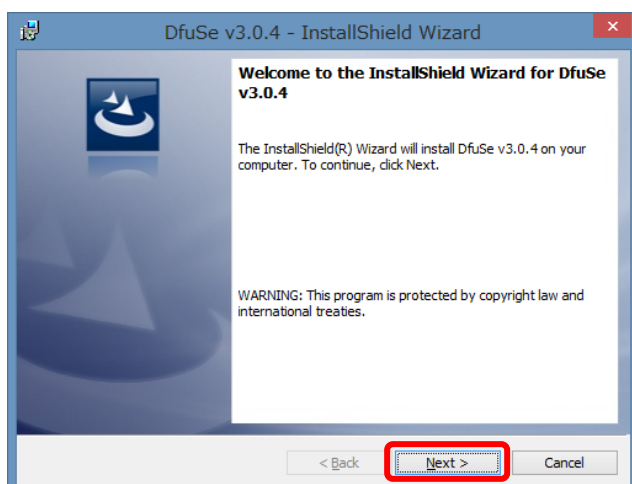
3. 【CoronIDE3】→【dfu】→【drivers】→【setup】→【DfuSe_Demo_V3.0.4_Setup.exe】
を実行して下さい。



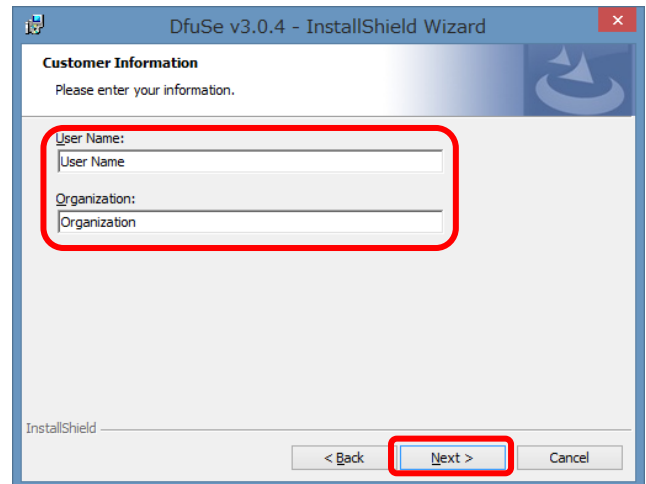
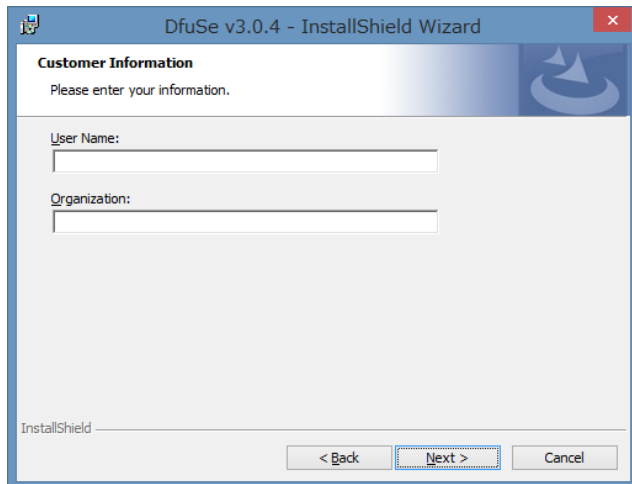
4. 「DfuSe」のインストーラーが起動します。



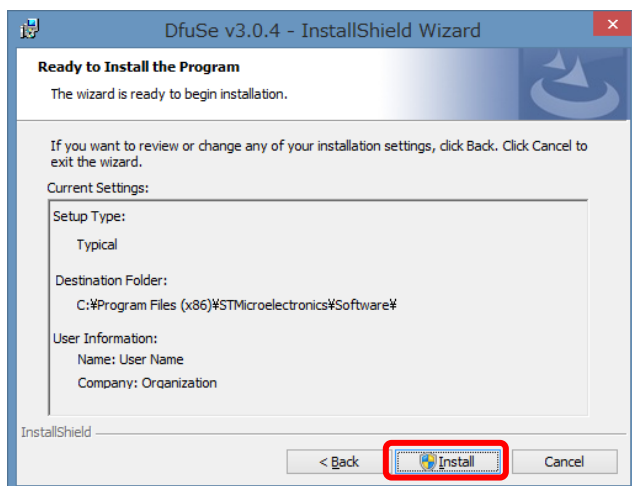
5. 【Next】を押してください。



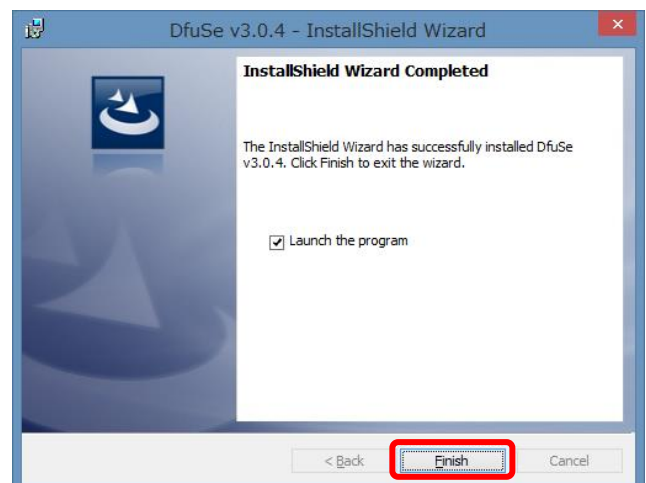
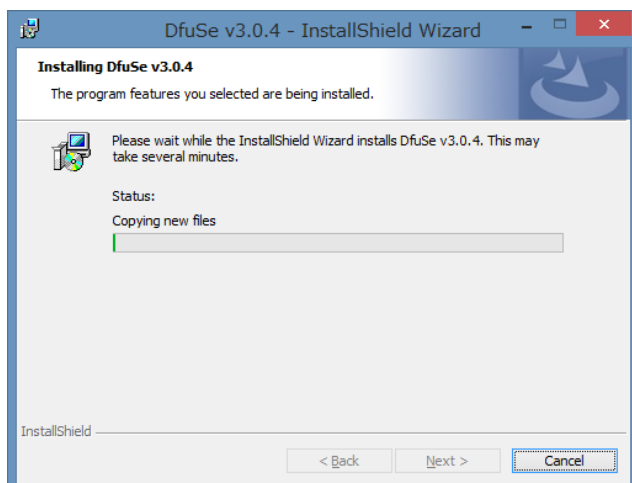
6. 【User Name】と【Organization】を入力（どちらも任意の文字で大丈夫です。）して、【Next】を押してください。



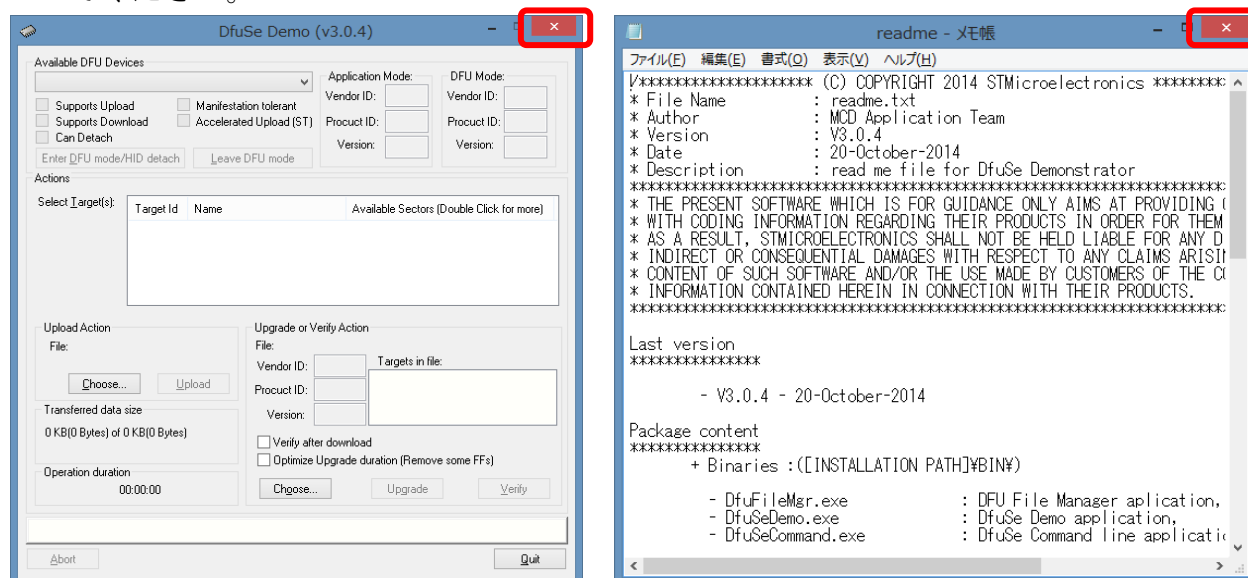
7. 【Install】を押してください。



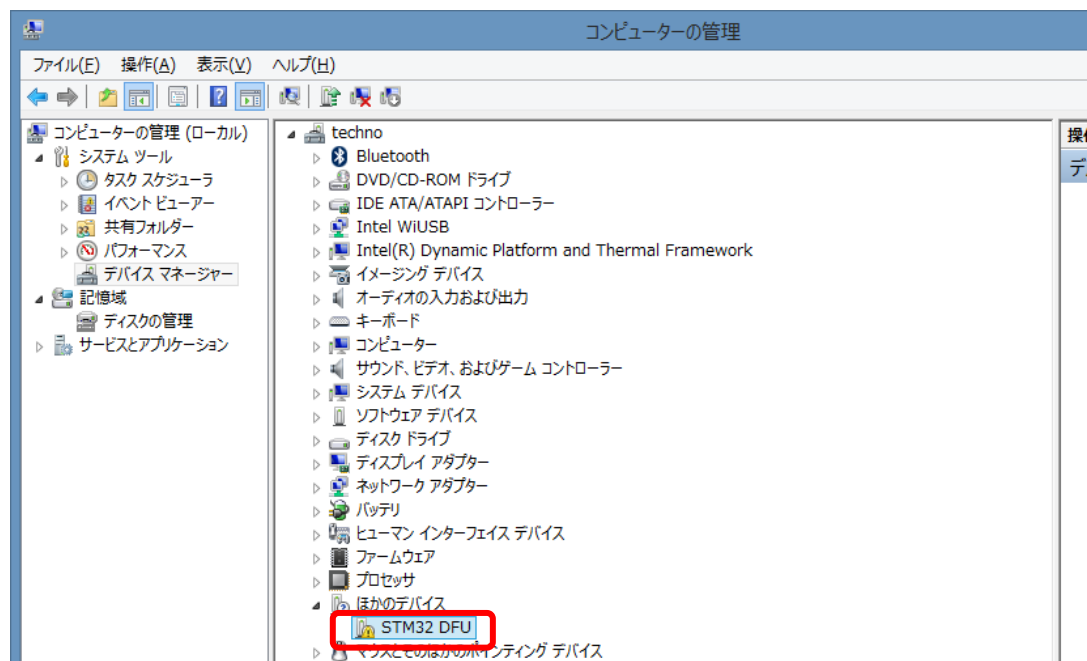
8. インストールが実行されます。完了したら、【Finish】を押して下さい。



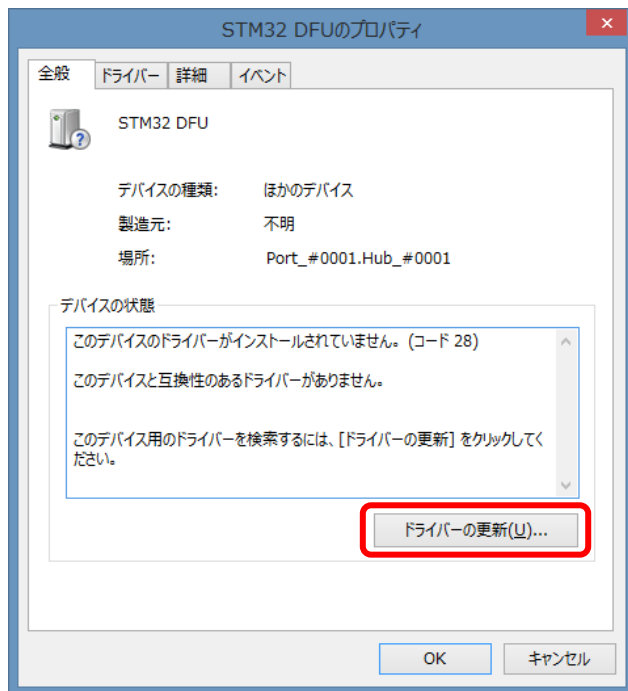
9. 「DfuSe Demo」が起動し、「readme」が表示されるので、両方とも×を押して終了してください。



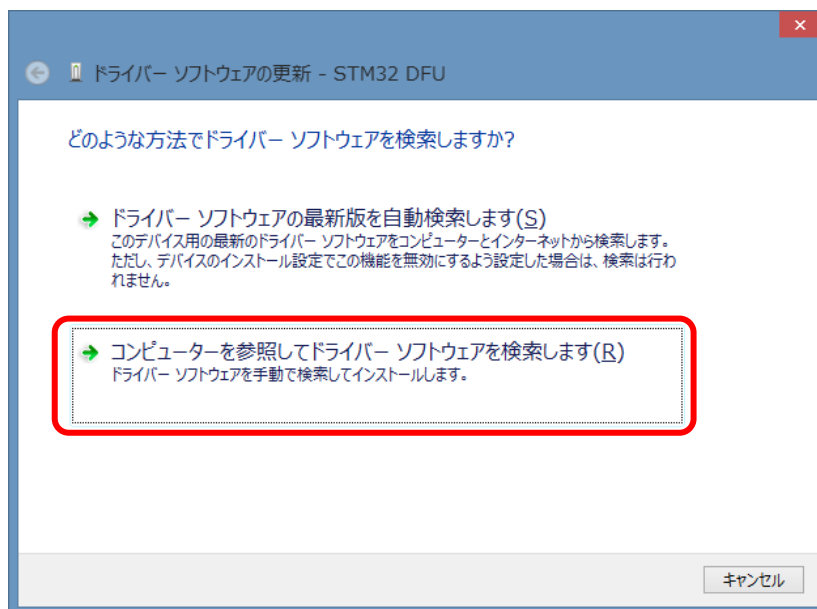
10. 【PC】→【管理】で「コンピューターの管理」を開き、【デバイスマネージャー】→【ほかのデバイス】→【STM32 DFU】を選択してください。



11. 【STM32 DFU】をダブルクリックすると下のウィンドウが表示されます。【ドライバーの更新】を選択してください。

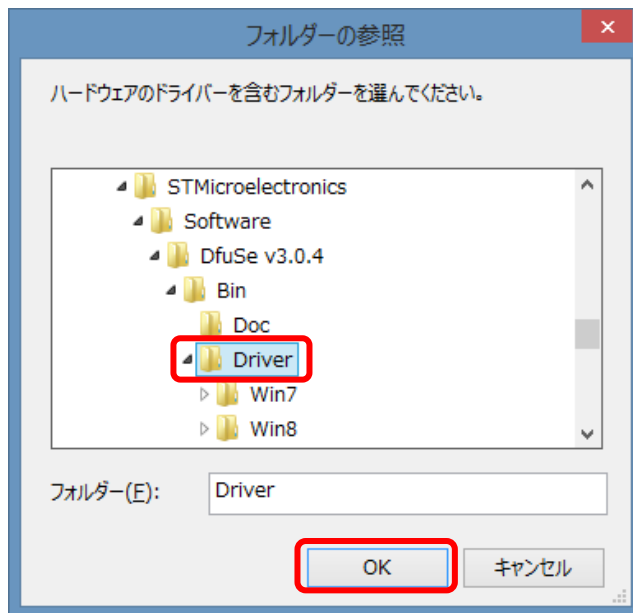


12. 「ドライバー ソフトウェアの更新 - STM32 DFU」が起動するので、【コンピューターを参照してドライバー ソフトウェアを検索します】を選択して下さい。

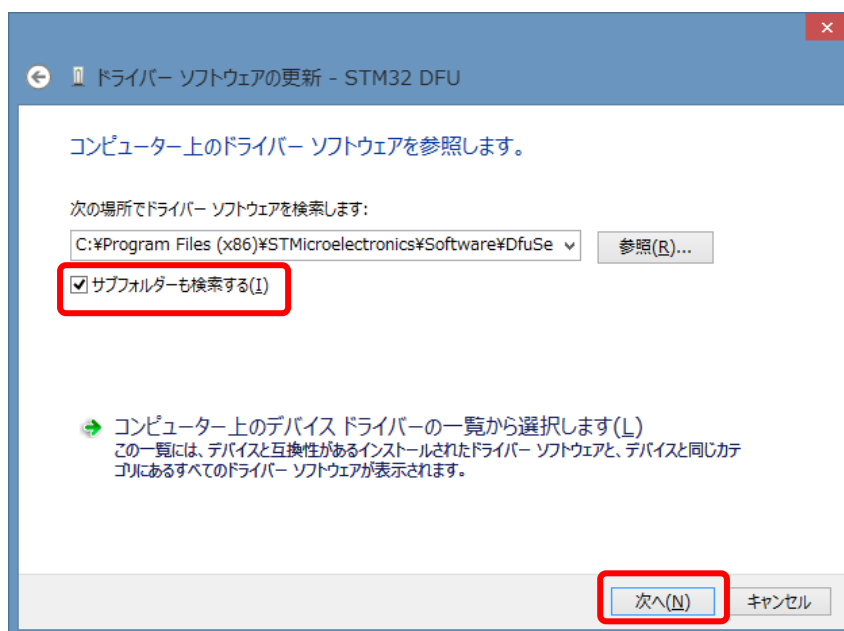


13. 【C:ドライブ】→【Program Files (x86)】→【STMicroelectronics】→【Software】→【DfuSe v3.0.4】→【Bin】→【Driver】を選択して下さい。

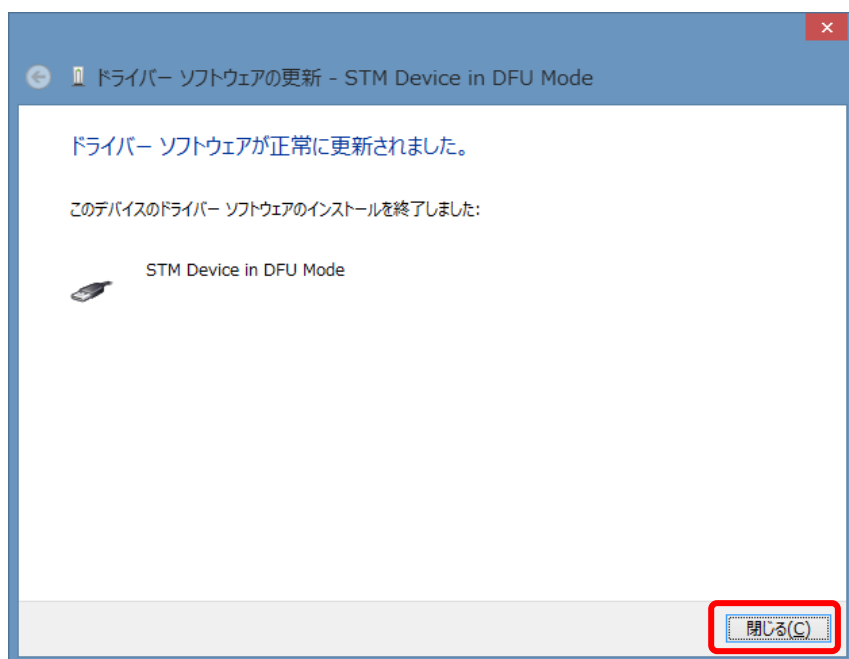
※デフォルト設定の 64bit 版 Windows7、8.1 でインストールした場合の保存場所です。
異なる場合はインストールフォルダ内を検索してください。



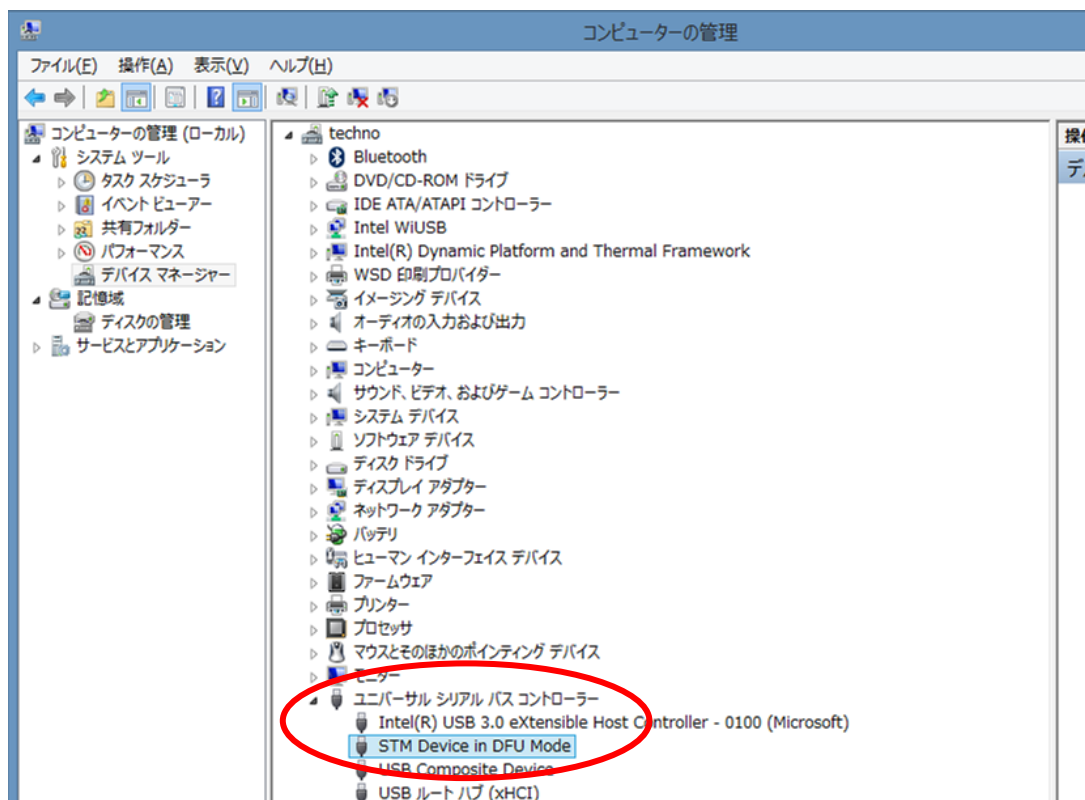
14. 【サブフォルダーも検索する】にチェックし、【次へ】を押して下さい。



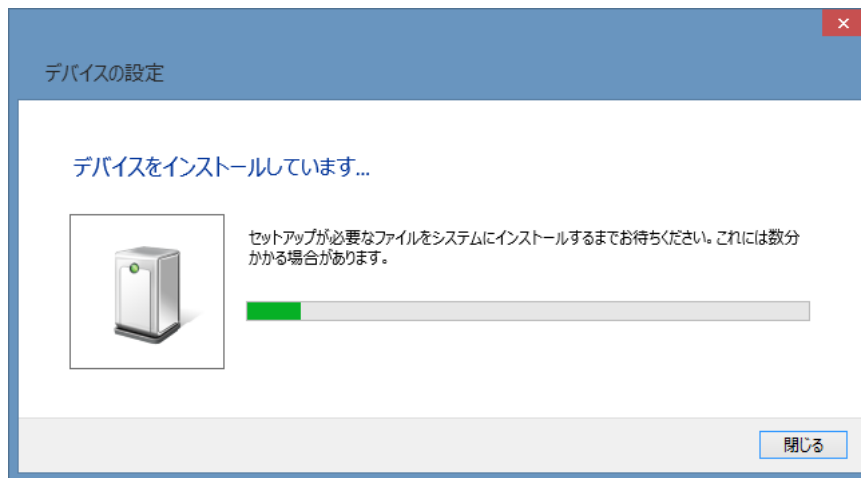
15. DFU のドライバーがインストールされます。インストールが終了したら、【閉じる】を押して下さい。



16. 【PC】 → 【管理】 で「コンピューターの管理」を開き、【デバイスマネージャー】 → 【ユニバーサル シリアル バス コントローラー】を選択し、そこに【STM Device in DFU Mode】と表示されていれば、DFU ドライバーのインストールが正常に完了し、プログラムを書き込める状態です。

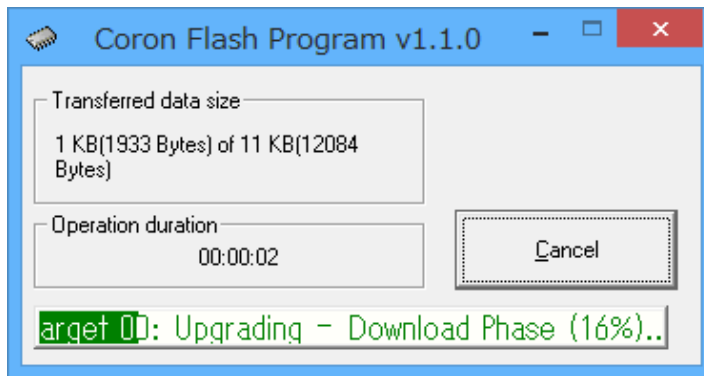


17. USB ポートを変えた場合、DFU のドライバーのインストールが必要になりますが、1 度インストール（手順 1～15）を行っていれば、自動的にドライバーがインストールされます。

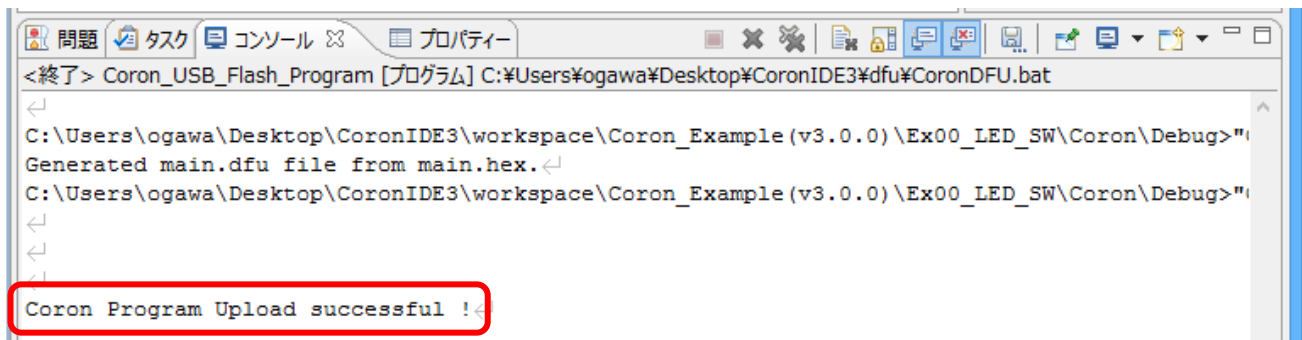


18. もし USB ポートを変えた時に、DFU のドライバーが自動的にインストールされなかった場合は、手順 10～15 を行って DFU のドライバーをインストールして下さい。

DFU ドライバーのインストールは以上です。次項からプログラムの書き込み方法を説明します。



Eclipse の「コンソール」タブに「Coron Program Upload successful!」と表示されたら、プログラムの書き込みが完了です。



Coron に電源を入れなおす（Coron から一旦 USB ケーブルを抜いて、何もスイッチを押さずに Coron と PC を USB 接続するか、もしくは 5～6V の AC アダプタを Coron と接続する）と、プログラムが始まります。

PC と Coron 間で USB 通信を行わないプロジェクトであれば、ケーブルを抜かずに Coron の SW_RESET（リセットボタン）を押せばプログラムが開始されるサンプルもあります。各サンプルのコメント文に従ってご使用下さい。

※書き込み時にエラーが表示された場合は下記のことを試してください。

【エラーメッセージ①】：

『‘Coron_USB_Flash_Program の起動’に問題が発生しました。
変数が空の選択を参照しています:\${project_loc}』

→画面左プロジェクト・エクスプローラーの現在開いているプロジェクト名のところで右クリック→【リフレッシュ】を実行。

【エラーメッセージ②】：

『No main.hex file.』
『This file doesn't have a correct format...』

→Eclipse 上メニューから【プロジェクト】→【クリーン】を選択して実行してください。

[エラーメッセージ③] :

『Coron in the DFU mode is not connected!

Please connect PC with Coron cabling USB while pushing SW_USER』

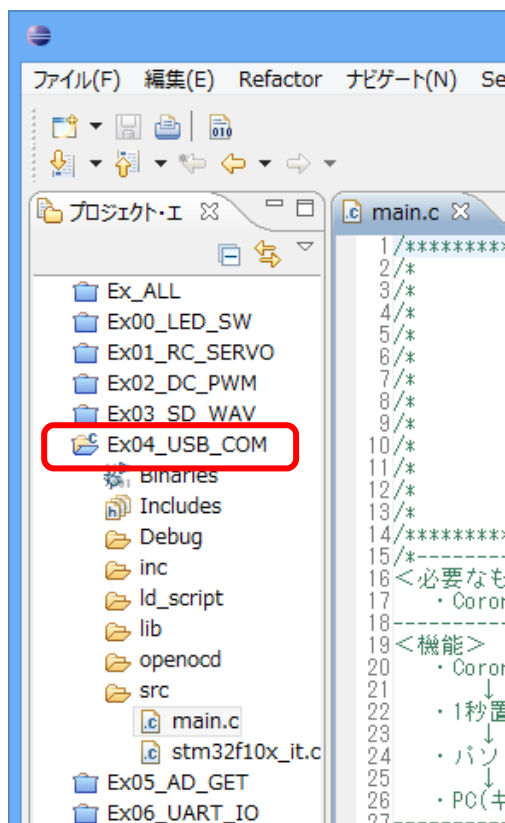
→DFU モードの Coron が認識できていません。もう一度、SW_USER を押しながら Coron の電源を入れなおしてください。

USB CDC ドライバーのインストール

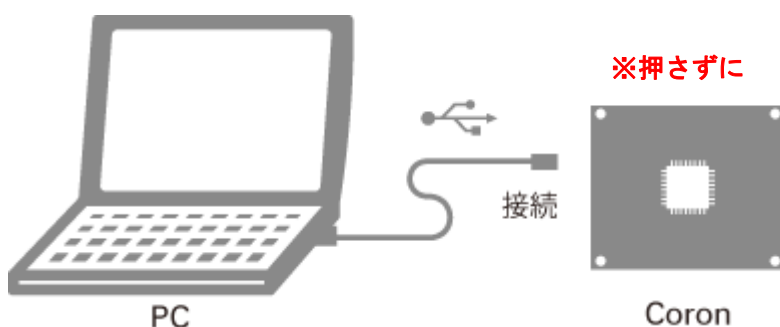
Coron は 2 つのドライバーをインストールする必要があります。

一つは先ほど記した DFU ドライバー、もう一つが Coron と PC 間で通信を行う際に必要となる USB ドライバーです。

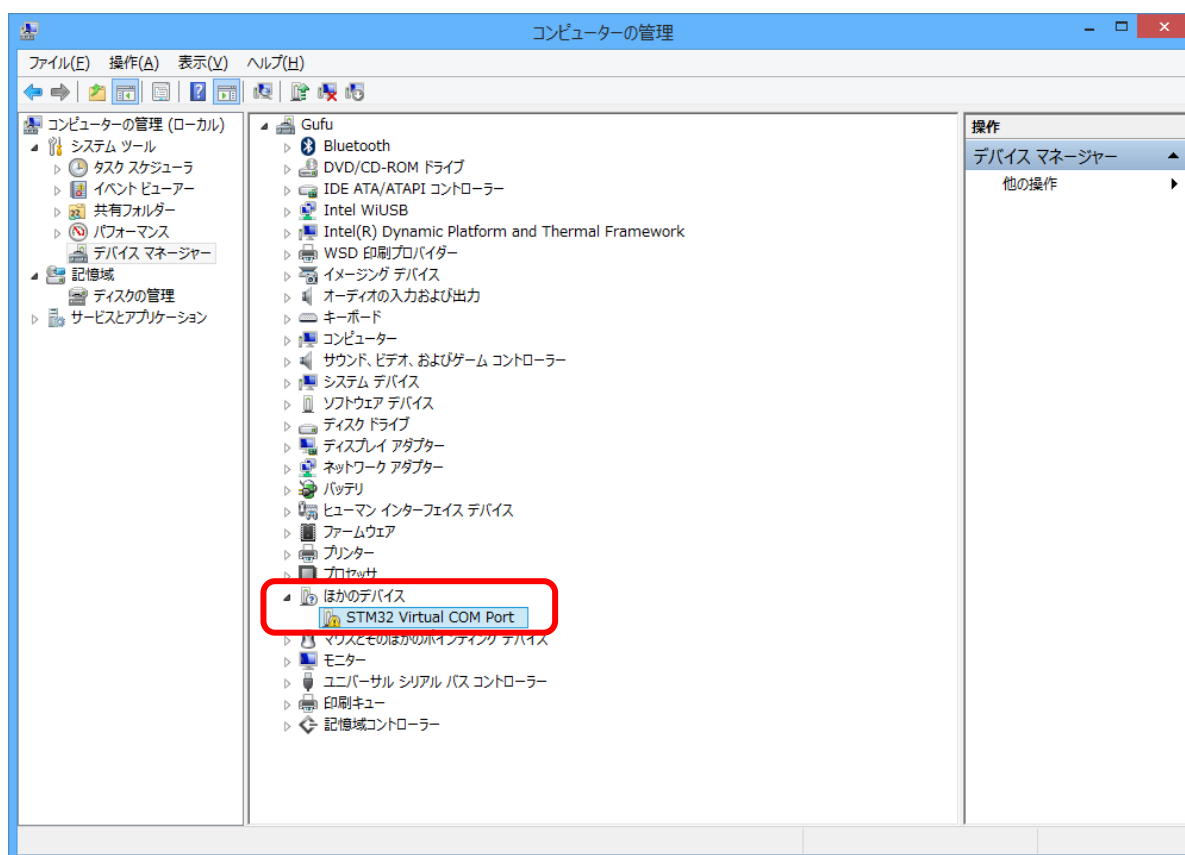
1. USB ドライバーのインストール方法を説明します。まず、Eclipse の「Ex04_USB_COM」というサンプルを開いてください。Coron を DFU モードで接続し、28 ページのプログラムの書き込みを参考に、Coron に「Ex04_USB_COM」を書きこんで下さい。



2. 「Ex04_USB_COM」が書きこめたら、いったん Coron から USB ケーブルを抜き、今度は何もスイッチを押さずに Coron を USB 接続してください。

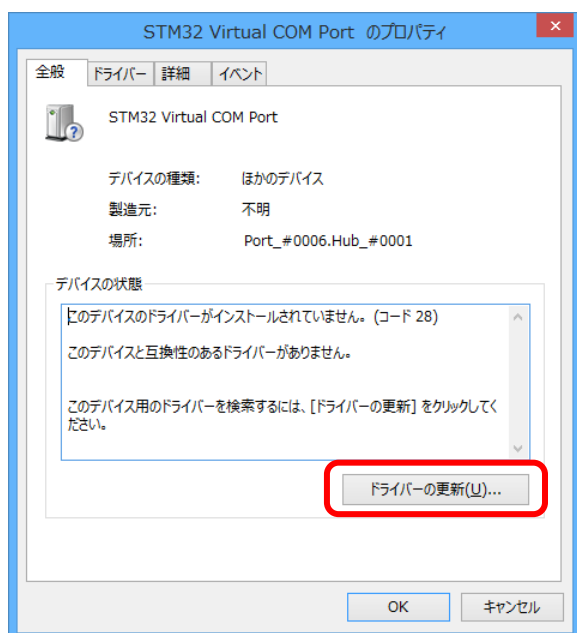


3. 【PC】→【管理】で「コンピューターの管理」を開き、【デバイスマネージャー】→【ほかのデバイス】を選択してください。

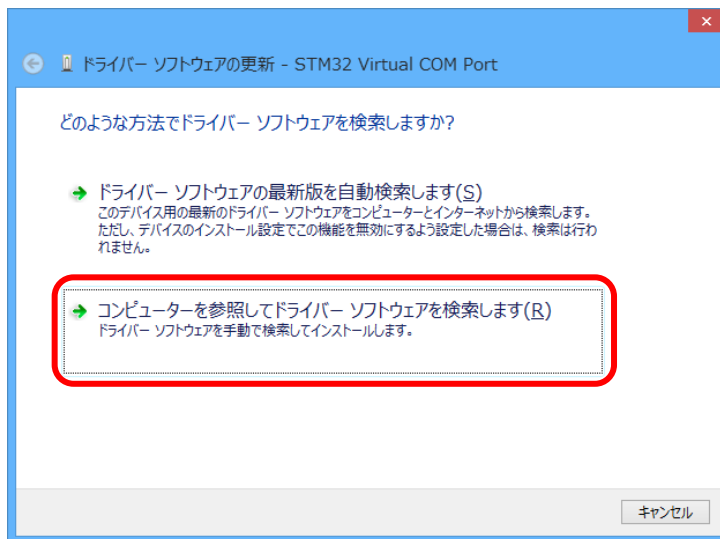


4. 【STM32 Virtual COM Port】を選択し、ダブルクリックして、「STM32 Virtual COM Portのプロパティ」を開いてください。

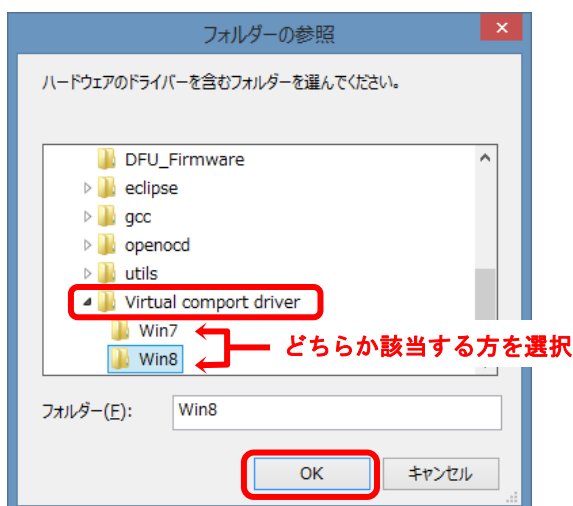
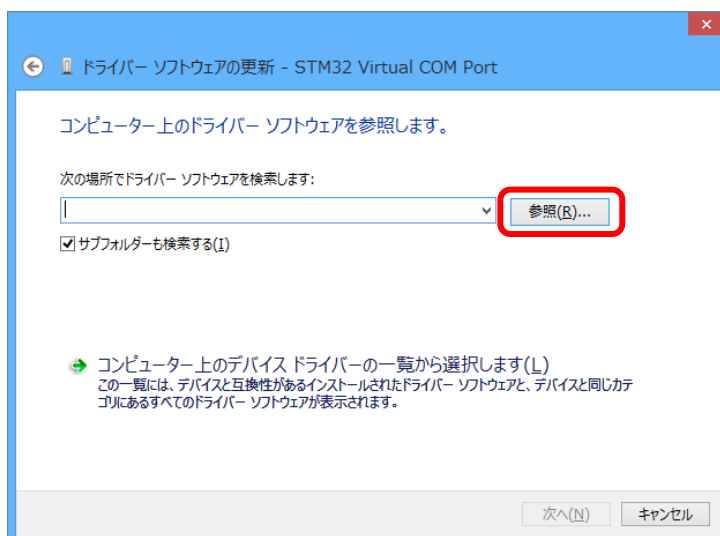
5. 【全般】→【ドライバーの更新】を選択し、「ドライバー ソフトウェアの更新 - STM32 Virtual COM Port」を開いてください。



6. 【コンピューターを参照してドライバー ソフトウェアを検索します】を選択して下さい。



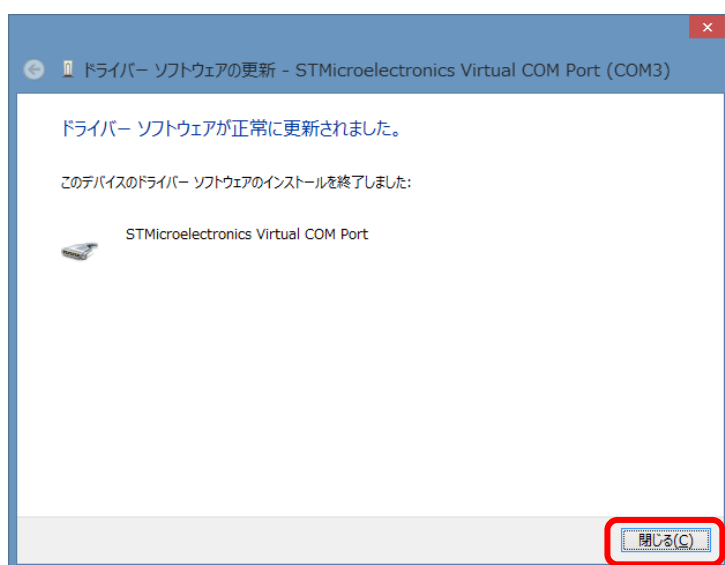
7. 【参照】をクリックし、【Virtual comport driver】→【Win7】、又は【Virtual comport driver】→【Win8】を選択して（お使いの OS に合わせて選択して下さい）【OK】をクリックします。



8. 【次へ】をクリックすると、ドライバーのインストーラーが起動します。インストーラーが起動したら【インストール】を選択します。（インストールをするか警告が出たら【はい】を選択しインストールを続行します）



9. インストールが完了したら【閉じる】をクリックして下さい。



10. Coron が接続されているポートを確認します。【PC】→【管理】で「コンピューターの管理」を開き、【デバイスマネージャー】→【ポート】で、【STMicroelectronics Virtual COM Port(COMx)】(x は数字) の「COMx」部分が Coron の接続先です。画像の例では【COM3】が接続先となっています。



以上で USB CDC ドライバーのインストールを含め、**Coron** のセットアップが完了しました。

あとは気になるサンプルを試したり、自由にプログラムを作って **Coron** をご活用下さい！

Coron へのプログラムの書き込み方法がわからなくなったら、28 ページのプログラムの書き込みをご確認下さい。

また、37 ページの Eclipse に新しくプロジェクトを追加するには、Eclipse に新しく **Coron** プロジェクトを作成する方法や、注意点などが記載されていますので、プロジェクトを新規作成したい場合はご確認下さい。

第4章 付属サンプルプログラムについて

Coron には多数のサンプルプログラムが用意されています。

全てのサンプルプログラムはそれぞれ Eclipse のプロジェクトとして構築されているので、そのまま Eclipse 上で展開ができます。

用意されているサンプルプログラム(プロジェクト)はそれぞれ以下の通りです。

※接続方法などの詳細は各フォルダ内の readme.txt を参照してください。

フォルダ名	プロジェクト名	主な機能	内容
Ex00_LED_SW	Coron_Ex00	LED、スイッチ	Coron ボード上の LED の点灯と プッシュスイッチによる LED 点滅
Ex01_RC_SERVO	Coron_Ex01	RC サーボ	RC1-0 に接続された RC サーボモータの 角度制御(時間制御の有/無)
Ex02_DC_PWM	Coron_Ex02	DC モータ	MOT1 に接続された DC モータの 速度制御(PWM 制御による加減速)
Ex03_SD_WAV	Coron_Ex03	SD カード、スピーカ	microSD カード内に保存した WAV ファイルの再生
Ex04_USB_COM	Coron_Ex04	USB	USB CDC クラス(仮想 COM ポート)を 使用してパソコンと通信
Ex05_AD_GET	Coron_Ex05	USB、AD コンバータ	Coron ボードに接続されたセンサを AD 変換してパソコンに出力
Ex06_UART_IO	Coron_Ex06	シリアル送受信	CN-UART(もしくは XBee)から シリアルデータの送受信
Ex07_LCD_CHAR	Coron_Ex07	拡張 I/O	CN-IO に接続されたキャラクタ 表示 LCD に文字を出力する
Ex08_LOG_TXT	Coron_Ex08	SD カード、AD コンバータ	タイマ割り込みで AD 変換して、 microSD カードに TXT ファイルでログ保存
Ex09_PWM_WAV	Coron_Ex09	SD カード、スピーカ、DC モータ	音楽再生しながら DC モータを駆動。 (EX03_SD_WAV と EX02_DC_PWM の融合)
Ex_ALL	Coron_ExALL	-	自作プログラミング用ひな型

※下記サイト、【Coron 掲示板】にて随時、最新ライブラリや Coron 情報を掲載していきますのでご覧下さい。

【Coron 掲示板】 URL : <http://techno-road.com/board.html>

Eclipse に新しくプロジェクトを追加する

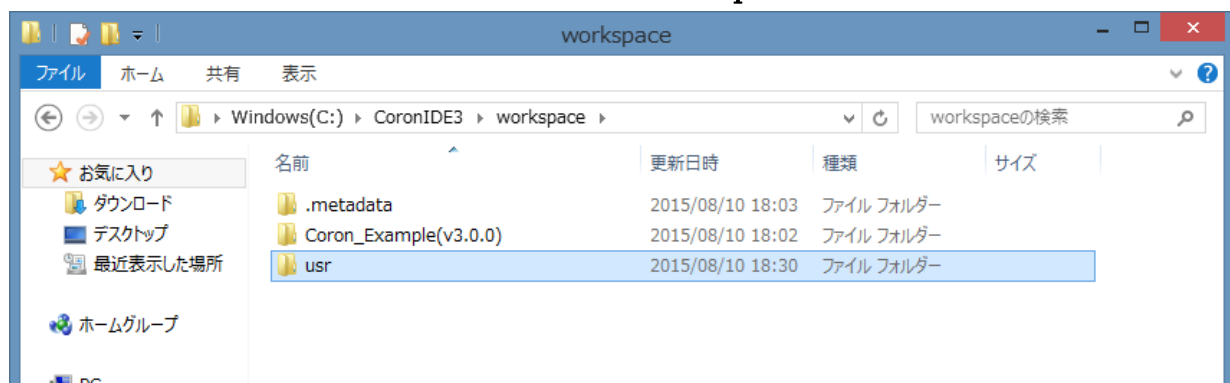
Eclipse にユーザが新しくプロジェクトを追加する方法を説明します。

ここでは、サンプルプロジェクト『Ex_ALL』をコピーしたものを例にしています。

- ① [Coron IDE3¥workspace] に自作プログラム作業用フォルダを作成します。

(※フォルダ名は任意です。ただし、スペースは使わず半角英数で作成して下さい。下図では「usr」)

Coron IDE3¥workspace



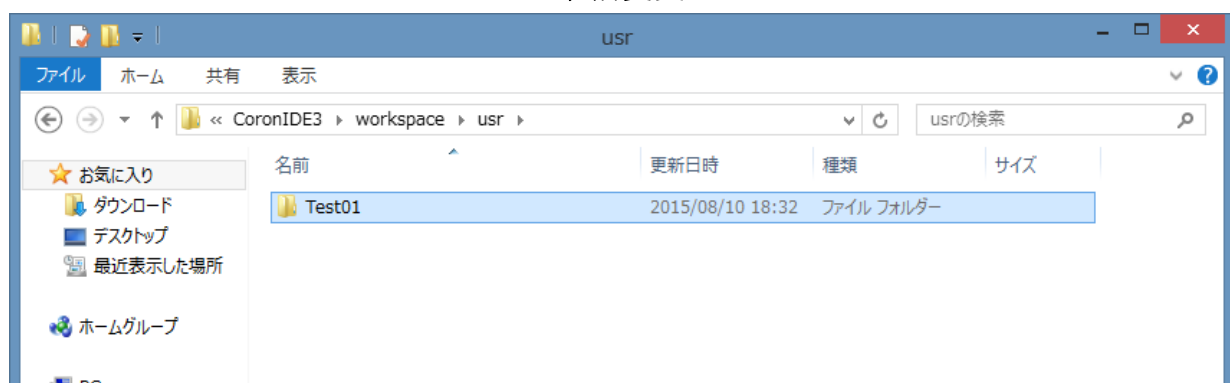
- ② ①で作ったフォルダに Ex_ALL をフォルダごとコピーします。

そしてフォルダ名を任意の名前に変更して下さい(※下図では「Test01」)

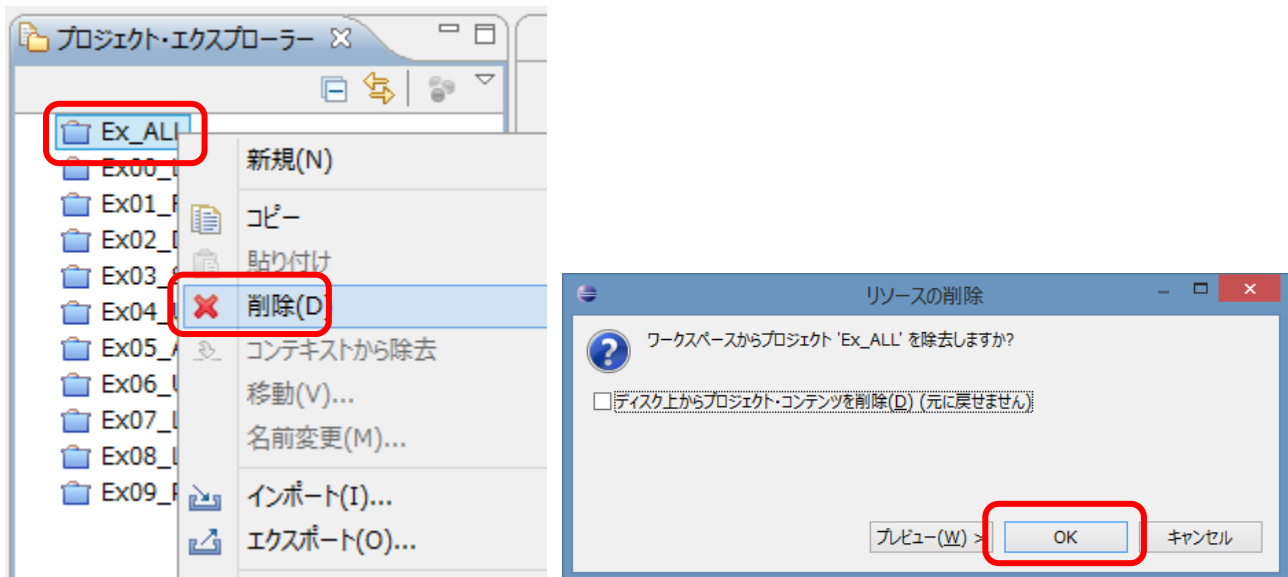
Coron IDE3¥workspace¥usr



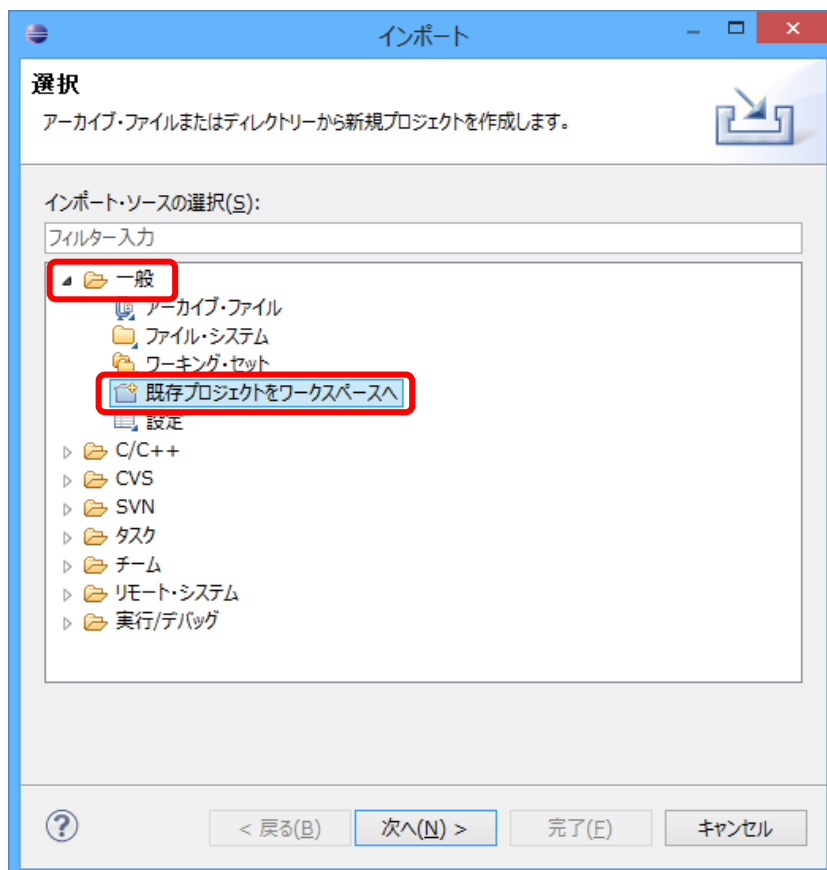
名前変更



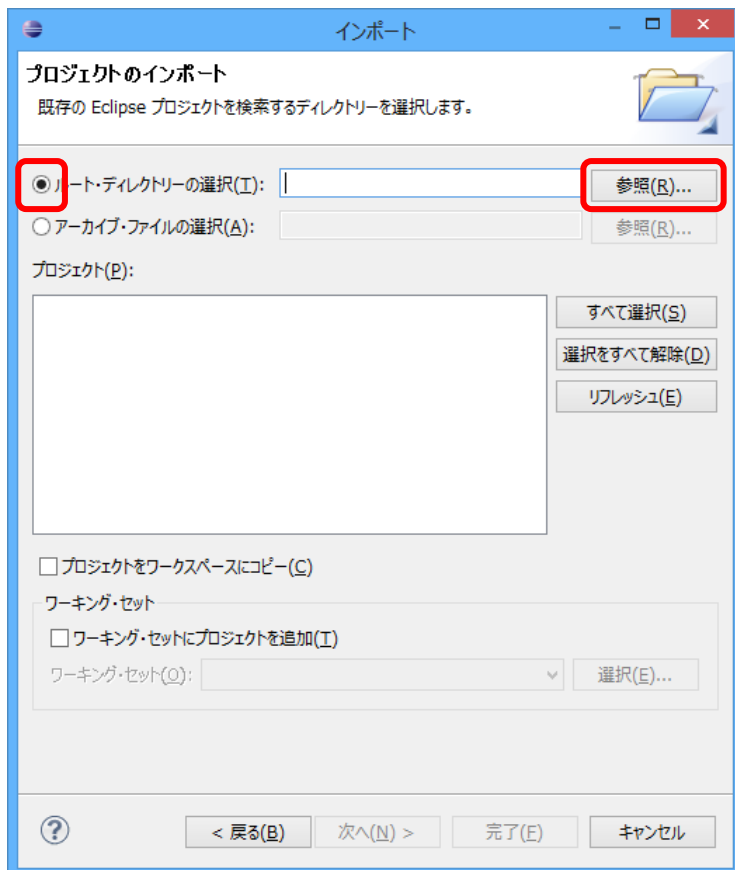
- ③ Eclipse を開いてプロジェクト・エクスプローラーから Ex_ALL を消去して下さい。
(※同じ名前のプロジェクトを2つは置けないため)



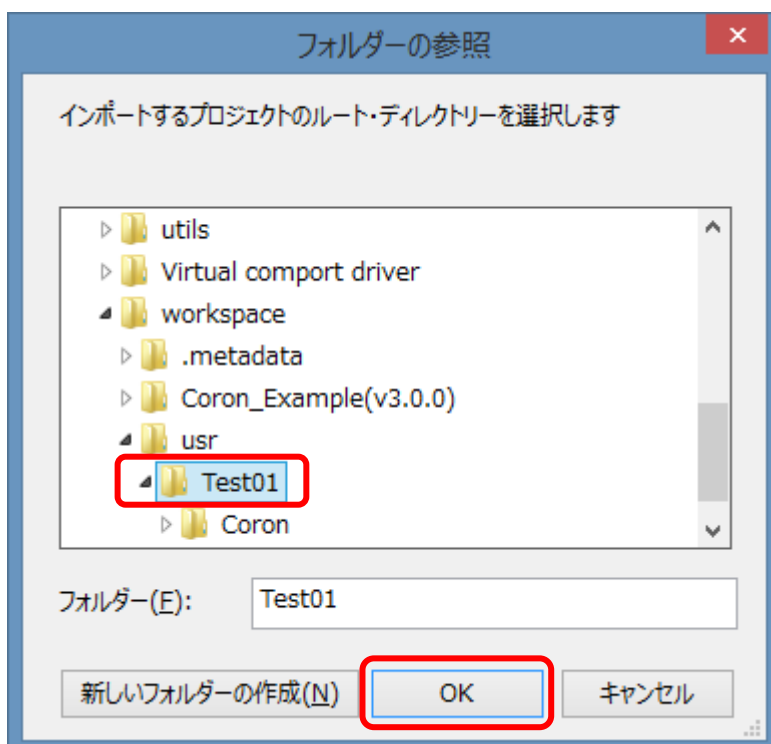
- ④ 【ファイル(F)】 → 【インポート(I)】 を選択し、プロジェクトをインポートします。
【既存プロジェクトをワークスペースへ】を選択して【次へ】を押してください。



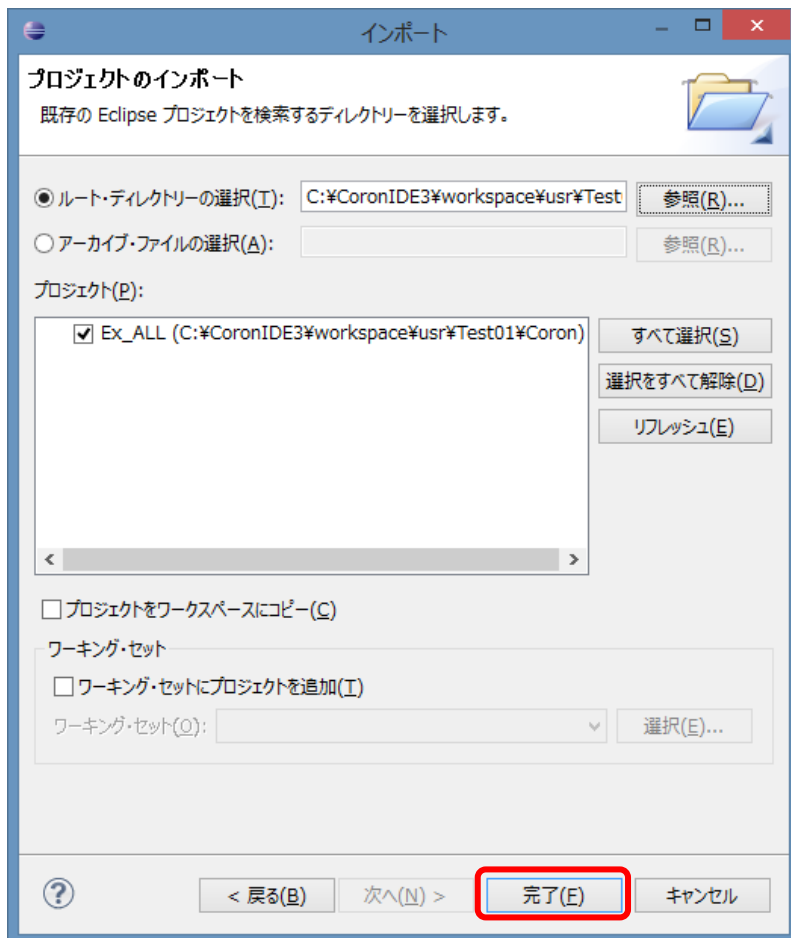
【ルート・ディレクトリーの選択】にチェックをし、【参照】を選択してください。



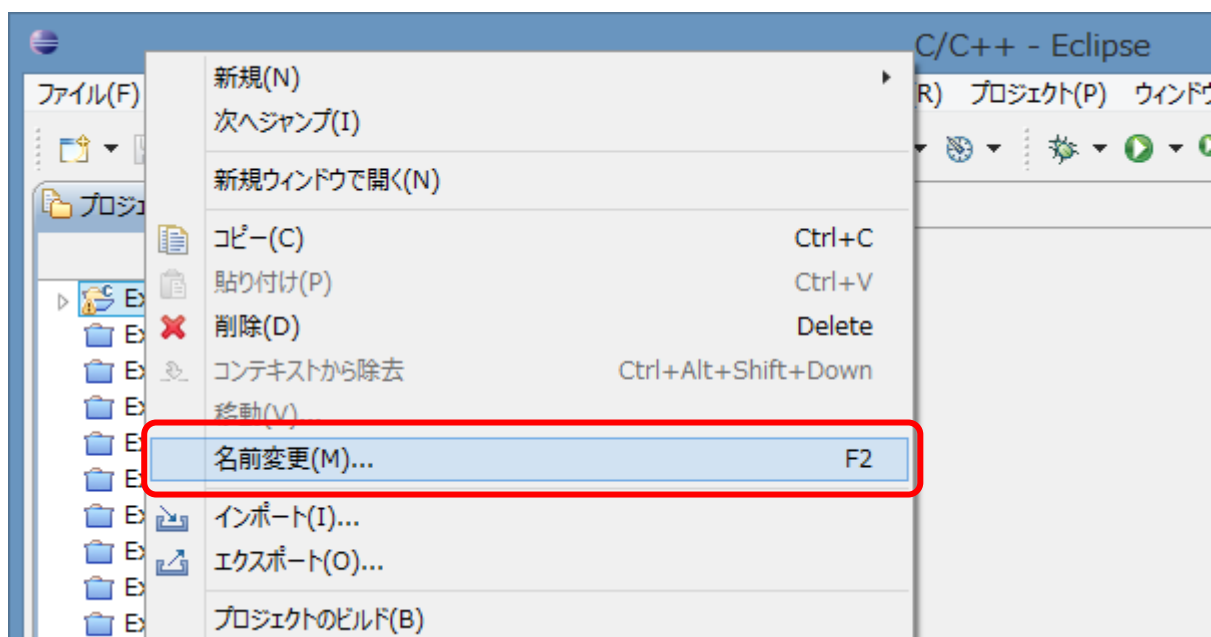
先ほどコピーしたフォルダを指定してください。
(※下図では [Coron IDE3¥workspace¥usr¥Test01])

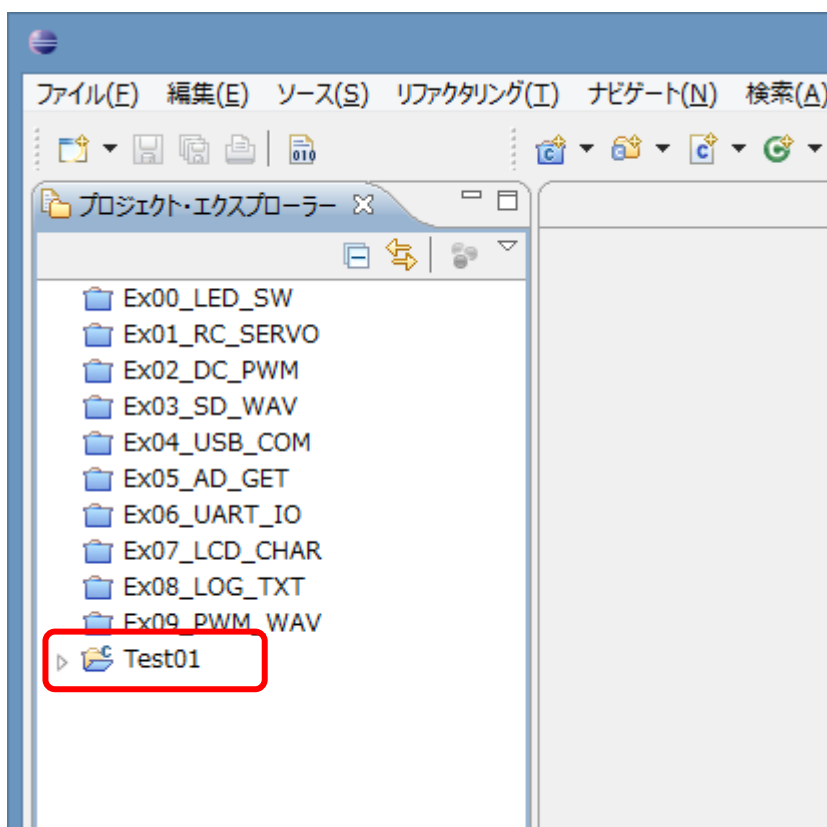
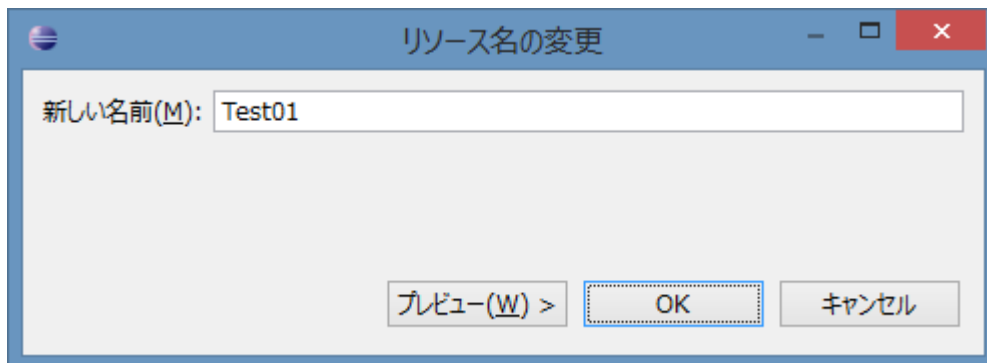
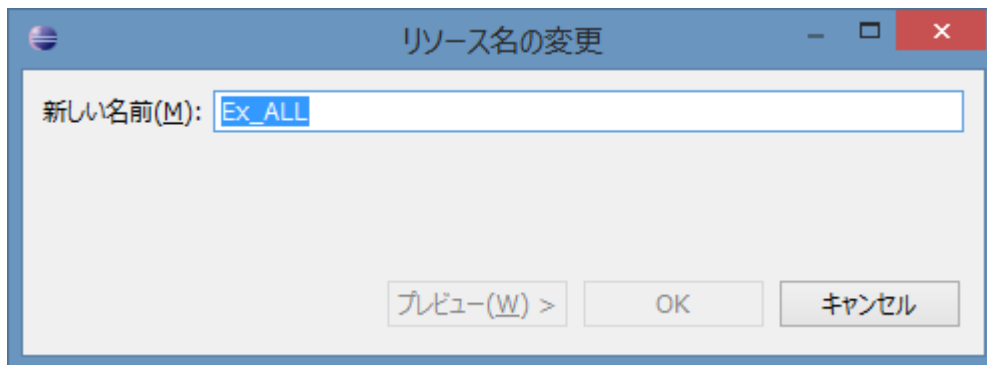


【完了】を選択します。



プロジェクト名【Coron_ExALL】を右クリックして、名前変更を選択します。
任意のプロジェクト名に変更して下さい。





これで「Ex_ALL」のプロジェクトコピーが出来たので、あとは「main.c」を自由に変更してユーザオリジナルのプログラムを作成してください。

※リソース名を変更した後は、コピー元の「EX_ALL」をインポートしなおしても問題ありません。

＜本文書について＞

本ドキュメントに含まれるソースプログラムおよび、本ドキュメントに関する権利や知的所有権は弊社が所有しています。

弊社の許可なく、本ドキュメントの全て、または一部に関わらず、複製、改変、転用等を行うことはできません。

＜商標について＞

本ドキュメントに掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

TECHNO ROAD Inc.

株式会社テクノロード

〒141-0022

東京都品川区東五反田 5-22-37

オフィスサークルN 五反田 303

<http://techno-road.com/>

E-mail:post@techno-road.com