

U16 向け ISP(イン・システム・プログラミング) サンプルプログラム AP ノート

ご注意

本資料の一部または全部をラピスセミコンダクタの許可なく、転載・複写することを堅くお断りします。

本資料の記載内容は改良などのため予告なく変更することがあります。

本資料に記載されている内容は製品のご紹介資料です。ご使用にあたりましては、別途仕様書を必ずご請求のうえ、ご確認ください。

本資料に記載されております応用回路例やその定数などの情報につきましては、本製品の標準的な動作や使い方を説明するものです。したがって、量産設計をされる場合には、外部諸条件を考慮していただきますようお願いいたします。

本資料に記載されております情報は、正確を期すため慎重に作成したものです。万が一、当該情報の誤り・誤植に起因する損害がお客様に生じた場合においても、ラピスセミコンダクタはその責任を負うものではありません。

本資料に記載されております技術情報は、製品の代表的動作および応用回路例などを示したものであり、ラピスセミコンダクタまたは他社の知的財産権その他のあらゆる権利について明示的にも黙示的にも、その実施または利用を許諾するものではありません。上記技術情報の使用に起因して紛争が発生した場合、ラピスセミコンダクタはその責任を負うものではありません。

本資料に掲載されております製品は、一般的な電子機器（AV 機器、OA 機器、通信機器、家電製品、アミューズメント機器など）への使用を意図しています。

本資料に掲載されております製品は、「耐放射線設計」はなされていません。

ラピスセミコンダクタは常に品質・信頼性の向上に取り組んでおりますが、種々の要因で故障することもあり得ます。

ラピスセミコンダクタ製品が故障した際、その影響により人身事故、火災損害等が起こらないようご使用機器でのディレーティング、冗長設計、延焼防止、フェイルセーフ等の安全確保をお願いします。定格を超えたご使用や使用上の注意書が守られていない場合、いかなる責任もラピスセミコンダクタは負うものではありません。

極めて高度な信頼性が要求され、その製品の故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのある機器・装置・システム（医療機器、輸送機器、航空宇宙機、原子力制御、燃料制御、各種安全装置など）へのご使用を意図して設計・製造されたものではありません。上記特定用途に使用された場合、いかなる責任もラピスセミコンダクタは負うものではありません。上記特定用途への使用を検討される際は、事前にローム営業窓口までご相談願います。

本資料に記載されております製品および技術のうち「外国為替及び外国貿易法」に該当する製品または技術を輸出する場合、または国外に提供する場合には、同法に基づく許可が必要です。

Copyright 2014 LAPIS Semiconductor Co., Ltd.

ラピスセミコンダクタ株式会社

〒222-8575 神奈川県横浜市港北区新横浜 2-4-8
<http://www.lapis-semi.com/jp/>

目次

1. はじめに.....	4
1.1. 関連資料.....	4
1.2. サポート MCU.....	4
2. ISP プログラム概要.....	5
2.1. ハードウェア構成.....	5
2.2. ソフトウェア構成.....	7
2.3. 通信パラメータと通信プロトコル仕様.....	10
2.4. メモリ配置.....	11
2.4.1. リマップ端子 'L' の場合.....	11
2.4.2. リマップ端子 'H' の場合.....	12
2.4.3. プログラムでリマップする場合.....	12
2.5. サンプルプログラムの動作.....	13
2.6. ソフトウェア構成図.....	14
2.6.1. ブートローダプログラム.....	14
2.6.2. ISP プログラム.....	14
2.7. サンプルプログラムを動かす.....	15
2.7.1. サンプルプログラムのビルド方法.....	15
2.7.2. サンプルプログラムの HEX 結合方法.....	16
2.7.3. サンプルプログラムの書き込み方法.....	18
2.7.4. バイナリファイル作成方法.....	20
2.7.5. XMODEM によるファイル転送方法.....	21
3. お客様のプログラムから ISP プログラムを起動する手順.....	23

1. はじめに

ISP(イン・システム・プログラミング)は、U16 内蔵フラッシュメモリのプログラムを書き換えるソフトウェアです。本 AP ノートは、ISP プログラムの概要と基本的な使用方法について説明します。

1.1. 関連資料

本 AP ノートに関連するマニュアル類を下記に示します。併せてご参照下さい。

- ・U16 向けリファレンスソフトウェア API マニュアル
- ・nX-U16/100 コア インストラクションマニュアル
- ・MACU8 アセンブラパッケージ ユーザーズマニュアル
- ・CCU8 ユーザーズマニュアル
- ・CCU8 プログラミングガイド
- ・CCU8 ランゲージリファレンス
- ・DTU8 ユーザーズマニュアル
- ・IDEU8 ユーザーズマニュアル
- ・uEASE ユーザーズマニュアル
- ・FWuEASE フラッシュライター ホストプログラム ユーザーズマニュアル

1.2. サポート MCU

本システムは以下の MCU をサポートしています。

- ü ML620Q154/Q155/Q156
- ü ML620Q504

2. ISP プログラム概要

2.1. ハードウェア構成

本サンプルプログラムが動作対象とするハードウェアの構成を以下に示します。

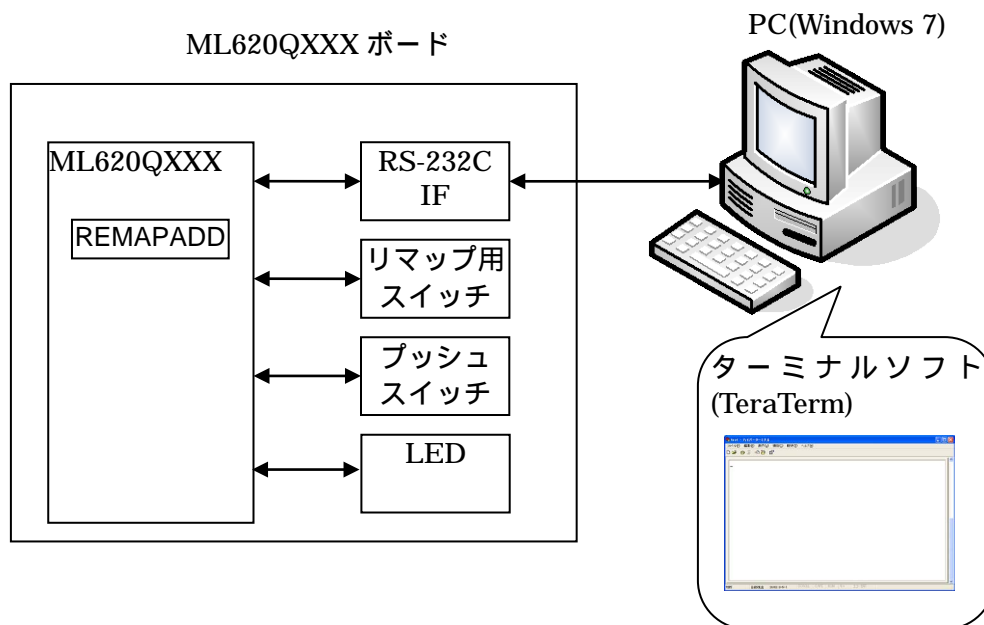


図 2-1 ハードウェア構成

表 2-1 MCU 周辺の仕様

項目	個数	詳細
REMAPADD	1 個	REMAPADD は、ソフトウェアリセット ¹ 後に、起動するアドレスを設定するためのレジスタです。
プッシュスイッチ (以下スイッチ)	1 個	プッシュスイッチは、サンプルプログラムが ISP プログラムを起動するために使用します。
RS-232C インターフェース	1 個	RS-232C インターフェースは、PC とのデータ送受信で使用します。
LED	1 個	LED は、エラー発生時に点灯します。
リマップ用スイッチ	1 本	リマップ端子は、ハードウェアリセット後に、ブートローダプログラムを起動するか、またはユーザプログラムを起動するかを切り替えるために使用します ² 。

*1 ソフトウェアリセットとは、特権レベルで BRK 命令を使用してソフトウェアで U16 コアをリセットすることです。

*2：リマップ用スイッチは、MCU に応じて、未実装の場合があります。対応表を以下に示します。

表 2-2 リマップ用スイッチ

MCU	リマップ用スイッチ	
	サポート	未サポート
ML620Q154/Q155/Q156		✓
ML620Q504	✓	

表 2-3 使用端子(ML620Q154/Q155/Q156)

端子	詳細
P02	UART RXD
P43	UART TXD
P20	LED
P00	プッシュスイッチ
*1	リマップ用スイッチ

*1 : ML620Q154/Q155/Q156 では、サポートしていません。

2.2. ソフトウェア構成

提供するサンプルプログラムは、以下のプログラムがあります。

表 2-4 サンプルプログラム

No	プログラム名	内容
1	ブートローダプログラム	ブートローダプログラムは、ISP プログラムを起動します。
2	ISP プログラム	ISP プログラムは、UART 経由で受信したデータをフラッシュメモリ (ユーザ領域 00000H) へ書き込みます。
3	ユーザプログラム	ユーザプログラムは、ユーザ領域で動作するプログラムです。本システムでは、ユーザプログラムは、スイッチが押下されると、ISP プログラムを起動します。

パッケージのディレクトリ・ファイル構成を以下に示します。

U16_ISP

```
| - License_e.pdf          ソフトウェア使用許諾書（英）
| - License_j.pdf         ソフトウェア使用許諾書（和）
| - readme(E).txt        README ファイル（英）
| - readme(J).txt        README ファイル（和）
|
| - DOC                   ドキュメントフォルダ
|   | - FEXT_U16_ISP_FLASH_AP-01.pdf    ISP サンプルプログラム AP ノート（英）
|   | - FJXT_U16_ISP_FLASH_AP-01.pdf    ISP サンプルプログラム AP ノート（和）
|
| - SampleSoft            ブートローダ用フォルダ
|   | - BOOTLOADER
|     | - _output
|     |   | - _obj
|     | - main
|     |   | - main.c
|     |   | - vector_bootloader.h
|     | - bootloader_large.PID          プロジェクトファイル(LARGE モデル)
|     | - bootloader_amall.PID          プロジェクトファイル(SMALL モデル)
|
|   - common              共通フォルダ
|     | - mcu_large
|     |   | - mcu.h
|     | - mcu_small
|     |   | - mcu.h
|     | - isp.h
|     | - led.h
|     | - remap.c
|     | - remap.h
|     | - sys_clock.c
|     | - sys_clock.h
|     | - sys_uart.c
|     | - sys_uart.h
|
|   - driver              ドライバ用フォルダ
|     | - clock
|     |   | - clock_i.h
|     | - common
|     |   | - rdwr_reg.h
|     | - flash
|     |   | - flash.c
|     |   | - flash.h
|     | - irq
|     |   | - irq_i.h
|     | - tbc
|     |   | - tbc.c
|     |   | - tbc.h
|     | - uart
|     |   | - uart_common_i.h
|     |   | - uart0_i.c
|     |   | - uart0_i.h
|     | - wdt
|     |   | - wdt.c
|     |   | - wdt.h
```


- ISP	ISP プログラム用フォルダ
- _output	
- _obj	
- main	
- crc.c	
- crc.h	
- isp.c	
- main.c	
- main.h	
- vector_isp.h	
- xmodem.c	
- xmodem.h	
- isp_large.map	
- isp_large.PID	プロジェクトファイル(LARGE モデル)
- isp_small.map	
- isp_small.PID	プロジェクトファイル(SMALL モデル)
- obj	オブジェクト用フォルダ
- isp_large.res	HTU8 用応答ファイル
- isp_small.res	HTU8 用応答ファイル(ML620Q156)
- ML620Q154_000RA.HEX	結合後の HEX
- ML620Q155_000RA.HEX	
- ML620Q156_000RA.HEX	
- ML620Q504_000RA.HEX	
- update_user_ML620Q154.bin	書き込み用バイナリファイル
- update_user_ML620Q155.bin	
- update_user_ML620Q156.bin	
- update_user_ML620Q504.bin	
- updated_user_small.res	
- SAMPLE_USER	サンプルユーザプログラム用フォルダ
- _output	
- _obj	
- main	
- main.c	
- main.h	
- vector_user.h	
- update_user_large.map	
- update_user_large.PID	
- update_user_small.map	
- update_user_small.PID	
- user_large.map	
- user_large.PID	
- user_small.map	
- user_small.PID	
- startup	スタートアップファイル用フォルダ
- ML620154.ASM	
- ML620155.ASM	
- ML620156.ASM	
- ML620504.ASM	
- ML620504_BOOTLOADER.ASM	
- readme.txt	

2.3. 通信パラメータと通信プロトコル仕様

PC との通信は、以下の通信パラメータおよび通信プロトコルを使用します。

表 2-5 通信パラメータ

ボーレート	115200bps
データ長	8bit
パリティ	なし
ストップビット	1bit

表 2-6 通信プロトコル

通信プロトコル	XMODEM
ブロックサイズ	128 バイト
エラー検出	CRC16

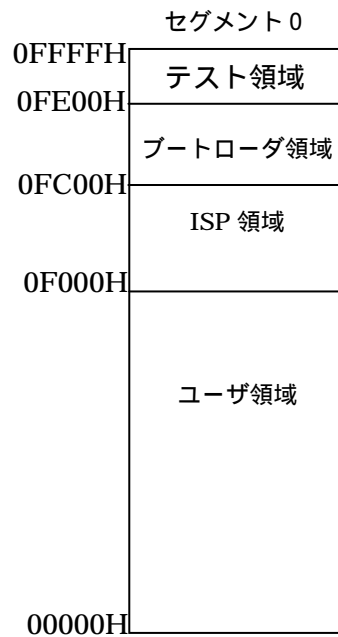
2.4. メモリ配置

2.4.1. リマップ端子 'L' の場合

リマップ端子を 'L' でハードウェアリセットをした時のメモリ配置は、以下の図のようになります。ただし、MCU によって ISP 領域の開始アドレスは異なります。各 MCU の ISP 領域の範囲を以下の表に示します。本 AP ノートは、ISP 領域の開始アドレスが 0F000H の場合について記載します。また、ブートローダ領域のサイズについては、使用する MCU のユーザズマニュアルを参照してください。

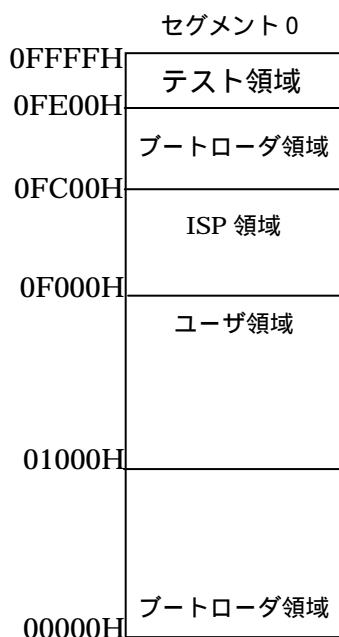
表 2-7 ISP 領域の範囲

MCU	ISP 領域の範囲	備考
ML620Q156 ML620Q504	0xF000 ~ 0xFBFF	
ML620Q155	0xB000 ~ 0xBBFF	
ML620Q154	0x7000 ~ 0x7BFF	



2.4.2. リマップ端子 'H' の場合

リマップ端子を 'H' でハードウェアリセットをした時のメモリ配置は、以下のようになります。



(注意)

以下の MCU は、リマップ端子がありません。

ML610Q154/Q155/Q156

2.4.3. プログラムでリマップする場合

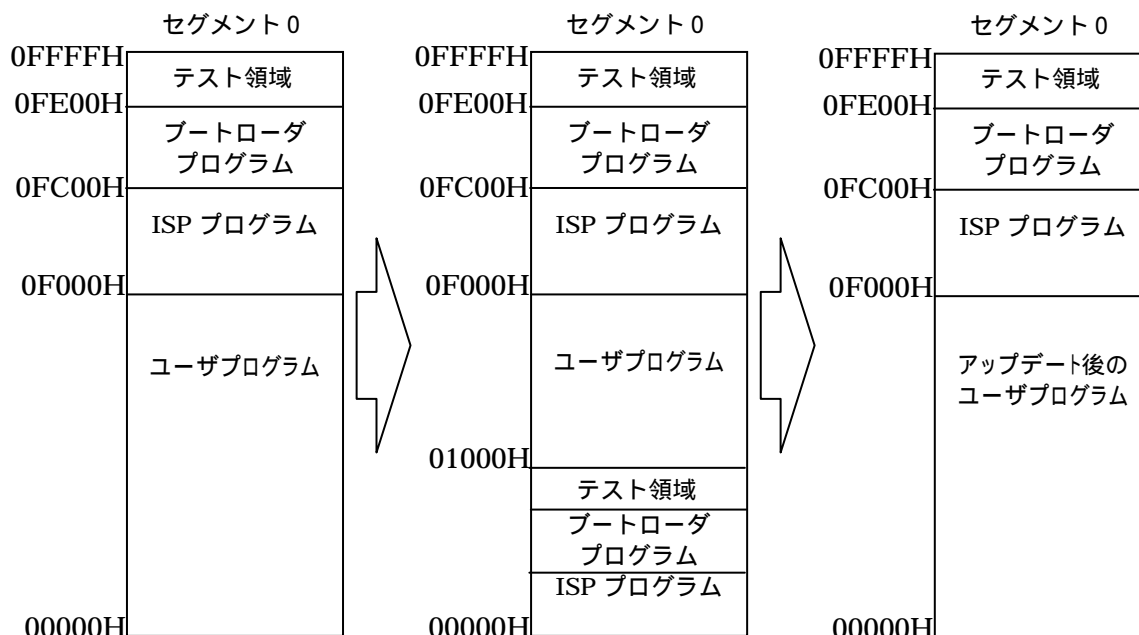
プログラムが REMAPADD レジスタを制御し、ソフトウェアリセットしてリマップする場合は、REMAPADD レジスタに指定したアドレス (4k バイト単位) が 0 番地から配置されます。

以下は、リマップアドレスを ISP 領域 (0F000H) と指定して、ソフトウェアリセットした場合のメモリ配置です。



2.5. サンプルプログラムの動作

本章では、サンプルプログラムを使用して、ユーザ領域のユーザプログラムを書換える動作について説明します。



- FWuEASE Flash Writer を使用して、ブートローダプログラム、ISP プログラム、ユーザプログラムをフラッシュメモリに書き込んでください。書き込み方法は、2.7.1 章から 2.7.3 章を参照してください。
 - ブートローダプログラムと ISP プログラム、ユーザプログラムを結合した HEX ファイルを以下に用意しています。
 - SampleSoft¥obj¥ML620QXXX_000RA.HEX (XXX は MCU 名)
- リマップ端子を“L”にして、ハードウェアリセットしてください。
 - 「ユーザプログラム」が起動し、ターミナルソフトに「Sample user program」が表示されます。(注意)
 - 以下の MCU は、リマップ端子がないため、ハードウェアリセット後すぐに「ユーザプログラム」が起動します。
 - ML620Q154/Q155/Q156
- スイッチを押してください。
 - 「ISP プログラム」が起動し、ターミナルソフトに「ISP」が表示されます。
- 「アップデート後のユーザプログラム」のバイナリファイルを PC から XMODEM プロトコルで送信してください。送信方法は、「2.7.5 XMODEM によるファイル転送方法」を参照してください。
 - 「アップデート後のユーザプログラム」として、以下のファイルを用意しています。
 - SampleSoft¥obj¥update_user_ML620QXXX.bin (XXX は MCU 名)
 - HEX をバイナリへ変換する方法は「2.7.4 バイナリファイル作成方法」を参照してください。
 - 以下のエラーが発生した場合、LED を点灯し、エラー状態(無限ループ)に入ります。
 - 通信エラー (受信タイムアウトを含む) によるリトライ回数が連続で 10 回を超えた
 - フラッシュメモリへの書換えエラー (ベリファイエラーもしくは MCU のフラッシュメモリサイズ以上の書き込み) が発生した場合
- 書換え終了後は、「アップデート後のユーザプログラム」が動作します。
 - アップデート後の「ユーザプログラム」が起動すると、ターミナルソフトに「Updated sample user program」が表示されます。

2.6. ソフトウェア構成図
2.6.1. ブートローダプログラム

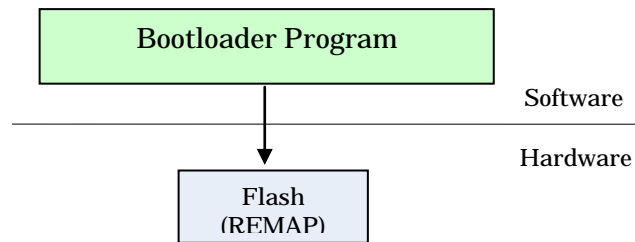


図 2-2 ブートローダプログラムのソフトウェア構成図

2.6.2. ISP プログラム

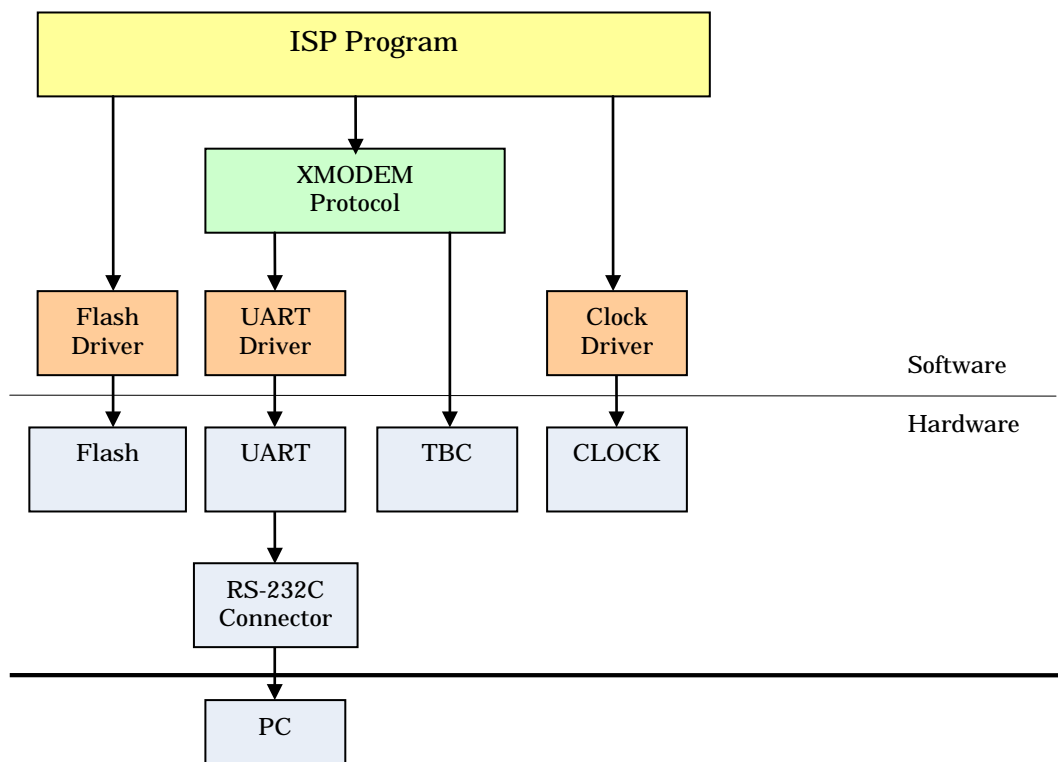


図 2-3 ISP プログラムのソフトウェア構成図

2.7. サンプルプログラムを動かす

2.7.1. サンプルプログラムのビルド方法

各サンプルプログラムに、プロジェクトファイルを用意しています。そのプロジェクトファイルを使用してビルドしてください。ビルドした結果、HEX ファイルが作成されます。この HEX ファイルをバイナリファイルに変換する方法は、「2.7.4 バイナリファイル作成方法」を参照してください。

表 2-8 プロジェクトファイル

プログラム名	プロジェクトファイル名
ブートローダプログラム	bootloader_small.PID bootloader_large.PID
ISP プログラム	isp_small.PID isp_large.PID
ユーザプログラム	user_small.PID user_large.PID
ユーザプログラム(アップデート用)	update_user_small.PID update_user_large.PID

1. IDEU8 を起動し、メニューから「開く」を選択し、使用する MCU のメモリモデルが SMALL モデルの場合は「XXX_small.PID」を、LARGE モデルの場合は「XXX_large.PID」を開きます。
 2. SMALL モデルの場合、デフォルトの設定では、ターゲット MCU は ML620Q156 が設定されています。異なる MCU をお使いの場合は以下の手順で設定を変更して下さい。
 - (1) IDEU8 のソースツリーウィンドウに登録されているスタートアップファイル ML620156.asm を削除し、代わりに、使用する MCU のスタートアップファイル(ML620QXXX の場合、ML620XXX.asm)を追加してください。
 - (2) メニューから「プロジェクト」→「オプション」→「コンパイル/アセンブル」を選択します。
 - (3) 表示されたウィンドウの「一般」タブ上の「ターゲット MCU」をお使いの MCU に設定します。
 - (4) MCU に対応するマクロを設定します。
「プロジェクト」→「オプション」→「コンパイル/アセンブル」→「マクロ」タブを選択して表示されるウィンドウで、以下の名前のマクロを定義して下さい。
 _ML620QXXX
 XXX の部分には、お使いの MCU の型番を対応させてください。
 (例)ML620Q156 を使用する場合、以下のマクロを定義してください。
 _ML620Q156
- 1.1 章に記載されているサポート MCU 以外の MCU をマクロで定義した場合、またメモリモデルが異なる PID ファイルを使用した場合はコンパイルエラーとなり、以下のようなエラーメッセージが出力されます。
- Error : E2000 : #error : “Unknown target MCU“
3. メニューから「プロジェクト」→「リビルド」を選択すると、サンプルプログラムのビルドが開始されます。
 4. ビルドが完了すると、プロジェクトファイルのあるフォルダに.abs ファイルが生成され、obj フォルダに.hex ファイルが生成されます。

2.7.2. サンプルプログラムの HEX 結合方法

HTU8 を使用し、3 つのプログラム(ブートローダプログラム、ISP プログラム、ユーザプログラム)の配置を指定して、1 つの HEX ファイルを作成します。

1. 以下のようにユーザプログラムを 0x0000 ~ 0xFFFFH、ISP プログラムを 0xF000H ~ 0xFBFFFH、ブートローダプログラムを 0xFC00 ~ 0xFDDFH、コードオプションデータを 0xFDE0 ~ 0xFDE1H に配置する場合は、以下のように応答ファイルを作成します。

```
//Response file(sample.res)

USER_SMALL.HEX /AL(0H, 0FFFFH, 0H) // ユーザプログラムの 0 ~ 0xFFFFH を 0H に配置

ISP_SMALL.HEX /AL(0H, 0BFFFH, 0F000H) // ISP プログラムの 0 ~ 0BFFFH を 0F000H に配置

//BOOTLOADER_SMALL.HEX /AL(0H, 01DFH, 0FC00H) // ブートローダプログラムの 0 ~ 01DFH を 0FC00H に配置

USER_SMALL.HEX /AL(0FDE0H, 0FDE1H, 0FDE0H) //コードオプションデータを 0FDE0H に配置
// (コードオプションデータを指定しない場合は不要)

/TML620156 // MCU 指定(ML620Q156 の場合は、/TML620Q156)。

/FML620Q156_000 // 出力ファイル名

/OH // 出力フォーマット指定(OH は Intel HEX フォーマット)
```

(注意)

ML620Q154/Q155/Q156 はリマップ端子がないため、上記例ではブートローダプログラムをコメントアウトしています。

応答ファイルの ~ については、MCU によってアドレスが異なりますので、以下のように変更してください。

表 2-9 各 MCU の対応アドレス

MCU	ML610Q156	0EFFFFH	0BFFFH	0F000H
	ML610Q504			
	ML620Q155	0AFFFFH		0B000H
	ML620Q154	06FFFFH		07000H

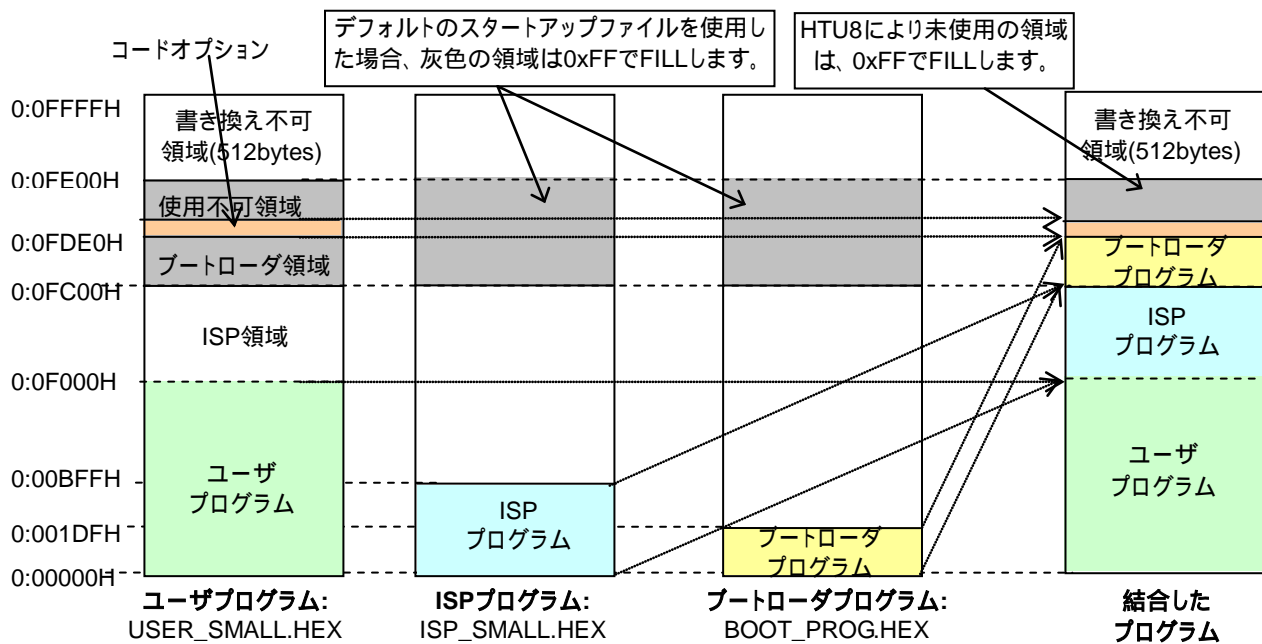


図 2-4 メモリマップ結合イメージ

2. [スタート] [すべてのプログラム] [U8 Tools] [nX-U8]から[Command-line 環境]を選択し、U8 用のコマンドプロンプトを開きます。
3. コマンドプロンプトの入力は以下のように入力します。

HTU8 @sample.res

変換に成功すると、HTU8 は ML620QXXX_000RA.hex と ML620QXXX_000RB.hex を生成します。2つのファイルは同じ内容ですので、どちらのファイルを使用してもかまいません。

2.7.3. サンプルプログラムの書き込み方法

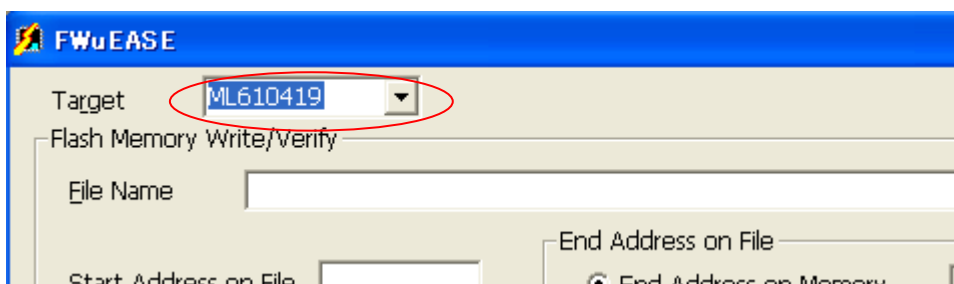
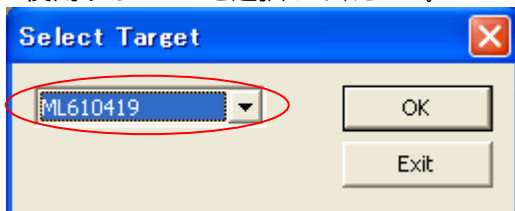
FWuEASE Flash Writer を使用してフラッシュメモリへ書き込みます。

以下に、「2.7.2 サンプルプログラムの HEX 結合方法」で作成した HEX ファイルを書き込む方法について説明します。

1. FWuEASE Flash Writer を起動します。

2. ターゲットの選択

使用する MCU を選択してください。



3. 書き込みファイル登録

(1) 2.7.2 章で作成した HEX ファイルを選択します。

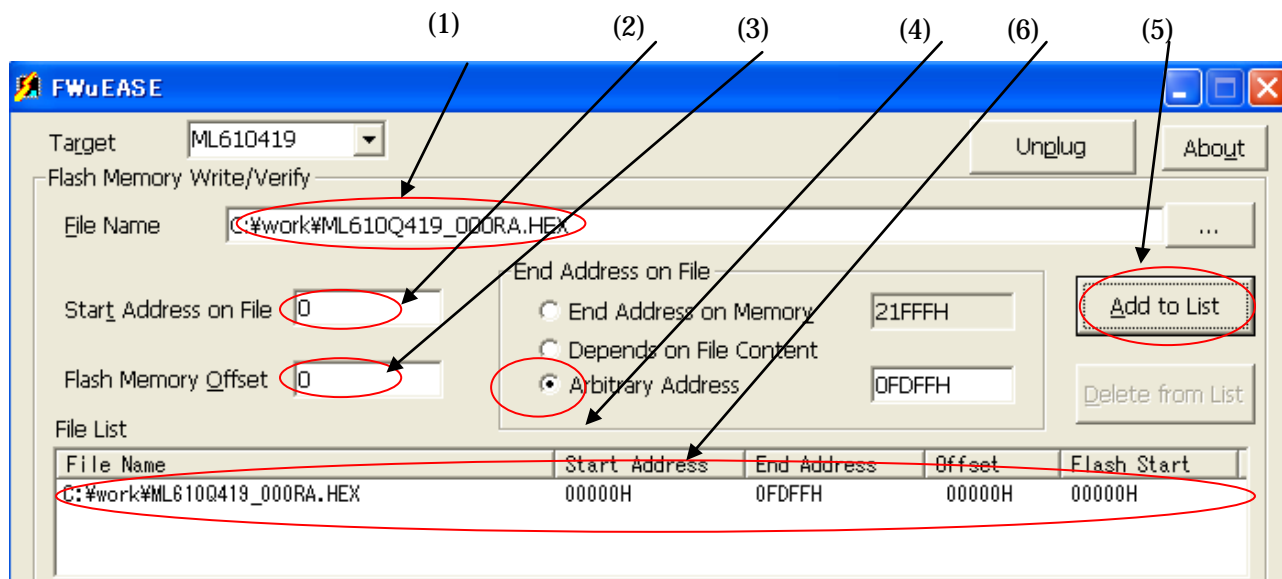
(2) [Start Address on File]に '0' を入力します。

(3) [Flash Memory Offset]に '0' を入力します。

(4) [End Address on File]の[Arbitrary Address]を選択し、'0FDFFH' を入力します。

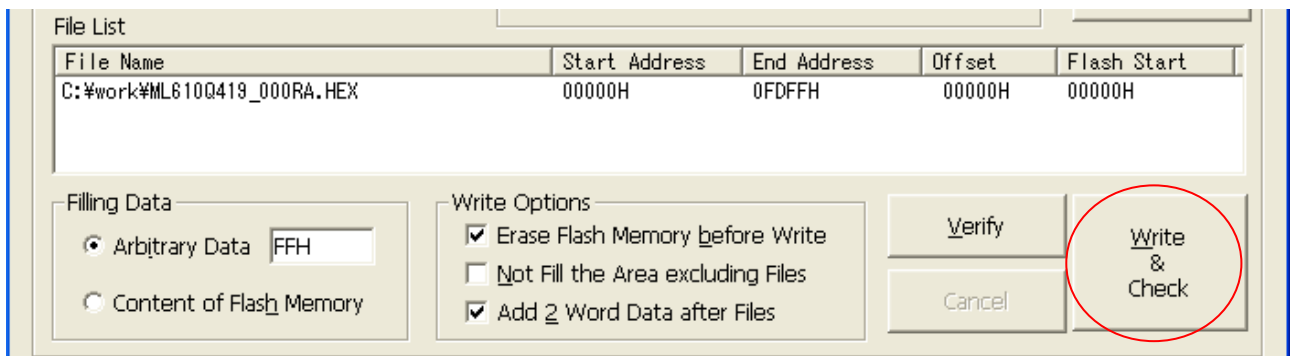
(5) [Add to List]ボタンをクリックします。

(6) [File List]に選択したファイルが登録されたことを確認します。

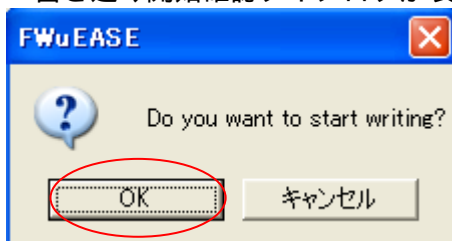


4. 書き込み開始

[Write&Check]ボタンをクリックします。

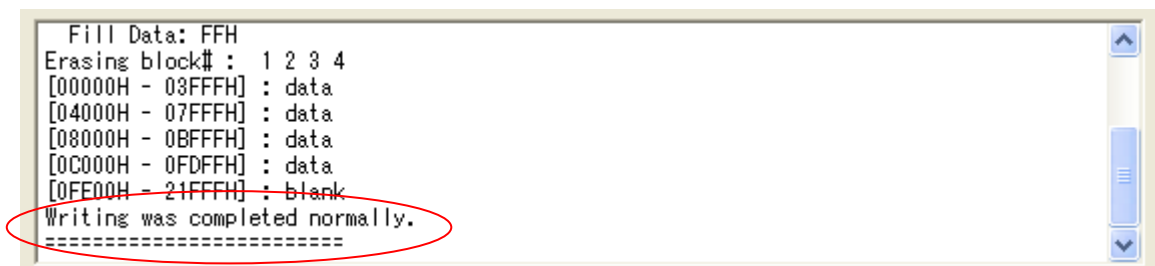


書き込み開始確認ダイアログが表示されますので、[OK]ボタンをクリックします。



5. 書き込み終了

正常に書き込まれると、以下のように“Writing was completed normally.”が表示されます。



2.7.4. バイナリファイル作成方法

アップデートするプログラムの形式は、バイナリファイルのみ対応しています。以下に HEX ファイルからバイナリファイルへの変換について説明します。

本システムでは、アップデートしたことが分かるように、ターミナルソフトに表示する文字を変更した "update_user_small.PID" を用意しています。本章ではこのファイルを使用した方法を説明します。

1. HEX ファイルを出力

IDEU8 のメニューから[プロジェクト] [オプション] [ターゲット]を選択します。一般タブの[HEX ファイルを出力する]、[インテルHEX]をチェックして、プログラムをビルドすると HEX ファイルが生成されます。このとき、「デバッグ情報を出力する」のチェックをはずしてください。

2. HEX ファイルのデータが書き込まれていない領域に 0FFFFH を埋め込む

[スタート] [すべてのプログラム] [U8 Tools] [nX-U8]から[Command-line 環境]を選択し、U8 用のコマンドプロンプトを開き、以下のように入力します。

```
HTU8 UPDATE_USER_SMALL.HEX /TML620156 /OH
```

変換に成功すると、HTU8 は UPDATE_USER_SMALLRA.hex と UPDATE_USER_SMALLRB.hex を生成します。2つのファイルは同じ内容ですので、どちらのファイルを使用してもかまいません。

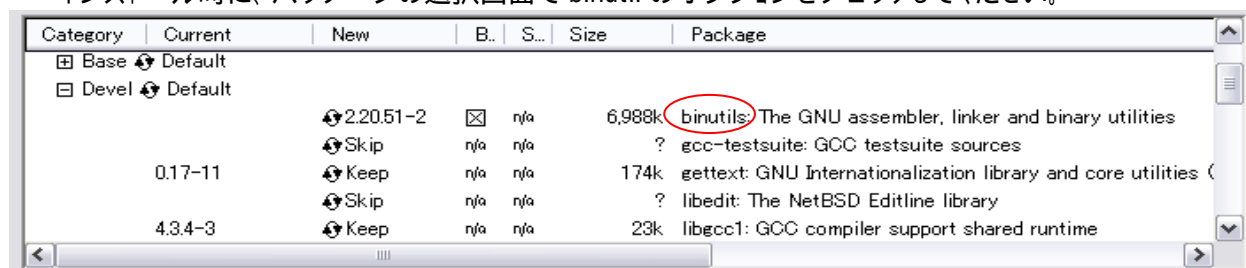
3. HEX ファイルからバイナリファイルに変換

ここでは cygwin 環境でバイナリファイルに変換する方法を紹介します。cygwin で以下のコマンドを入力すると HEX ファイルをバイナリファイルに変換します。

```
$ objcopy -lihex -Obinary UPDATE_USER_SMALLRA.hex UPDATE_USER_ML610Q156.bin
```

cygwin のインストーラは www.cygwin.com から入手できます。

インストール時に、パッケージの選択画面で binutils のオプションをチェックしてください。



(注意)

上記で作成したバイナリファイルは、2 で指定した MCU のフラッシュメモリの容量で作成されます。お使いの MCU のフラッシュ ROM 以降のデータは削除してください。

例)

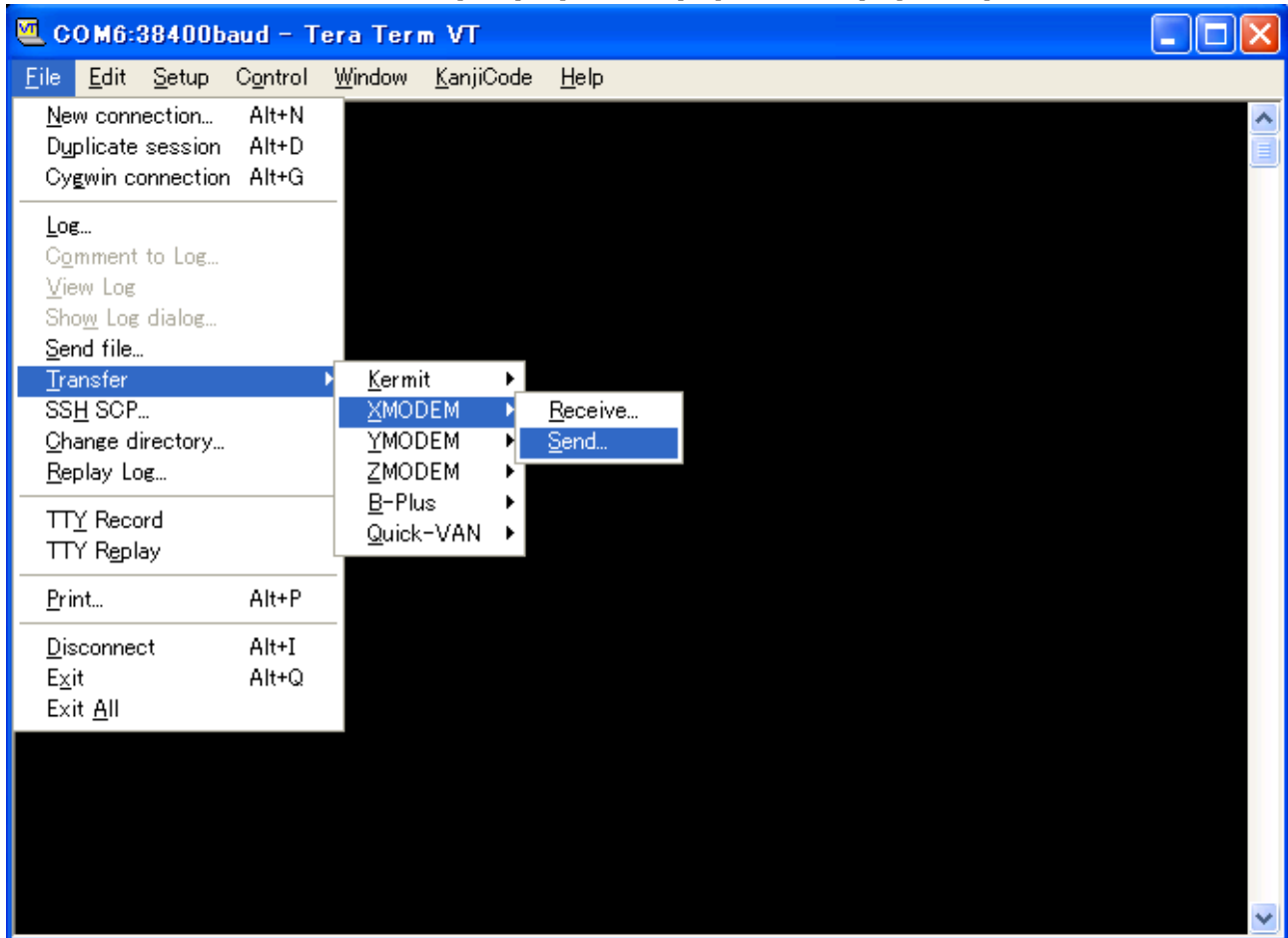
ML620Q156 をお使いの場合: MCU の ROM サイズは 64KB なので、64KB 以降は削除する。

2.7.5. XMODEM によるファイル転送方法

本システムは、XMODEM プロトコルを使用してファイル転送します。
Tera Term を使用する場合は、以下に説明します。

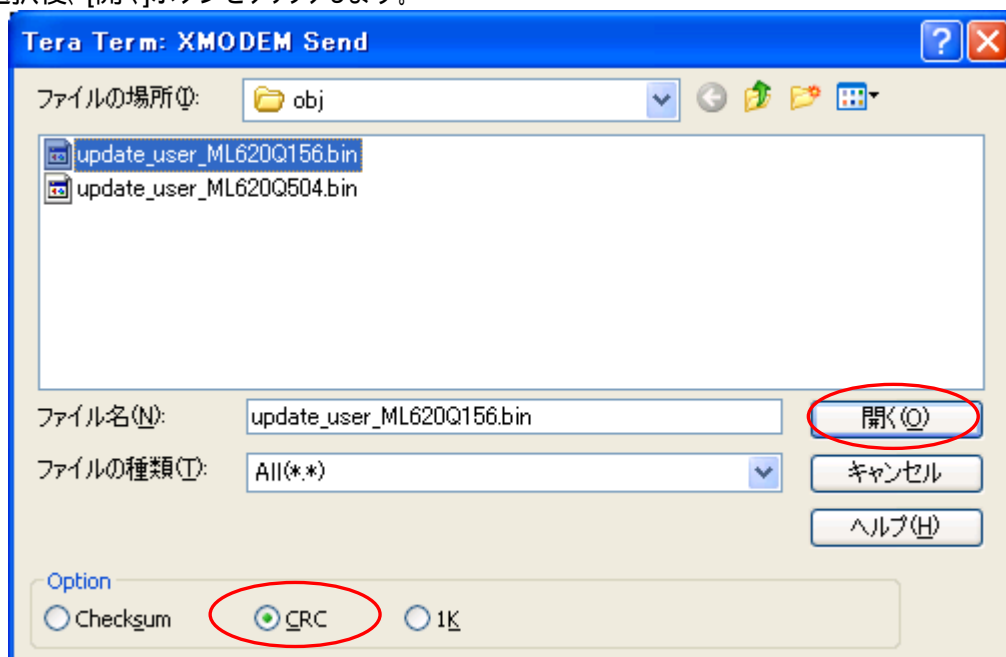
1. XMODEM プロトコルのファイル送信選択

Tera Term 起動後、メニューから [File] -> [Transfer] -> [XMODEM] -> [Send..] を選択します。



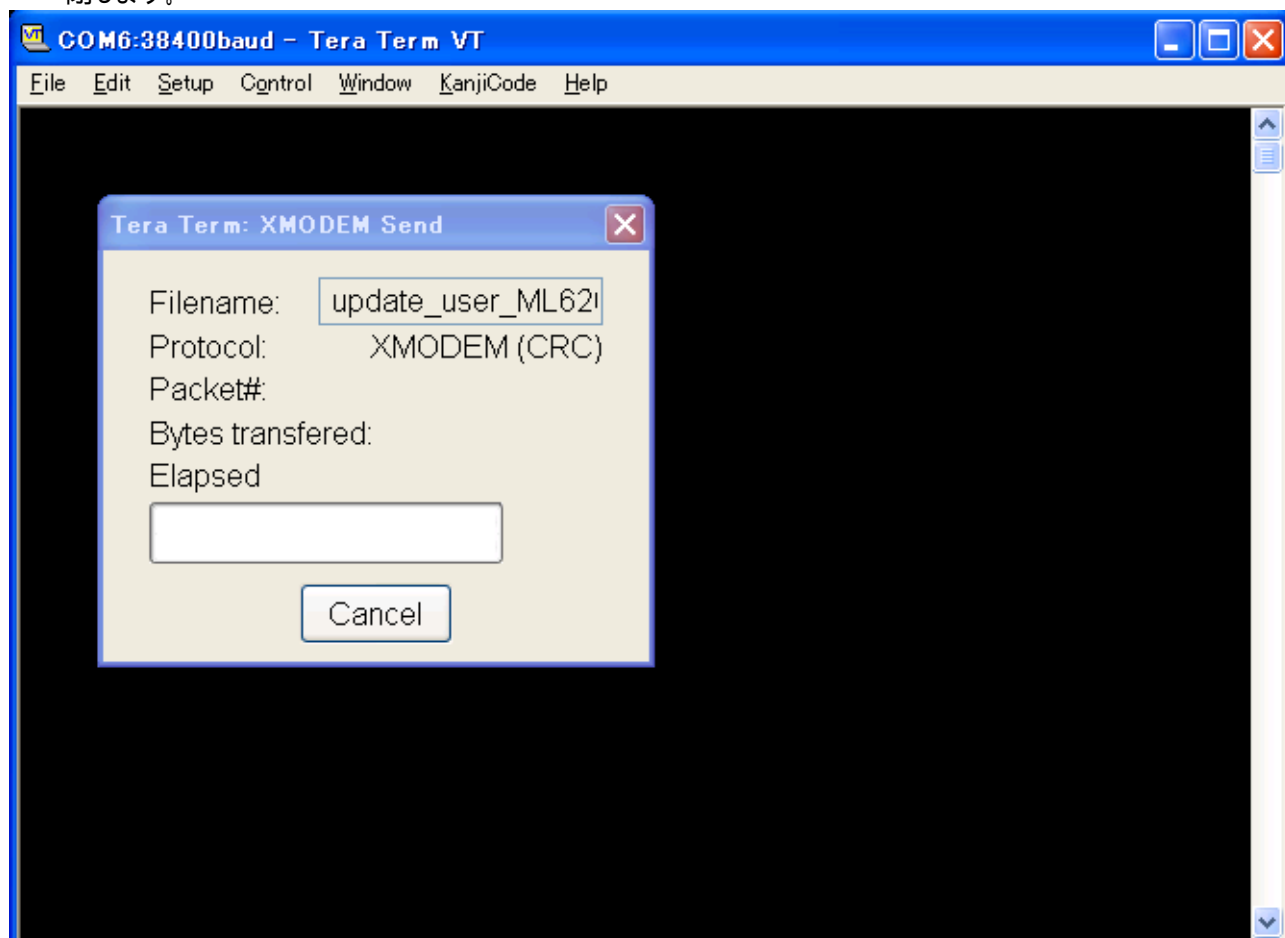
2. 送信ファイル指定とプロトコル選択

送信するファイルを選択します。Option で「CRC」を選択します。
上記選択後、[開く]ボタンをクリックします。



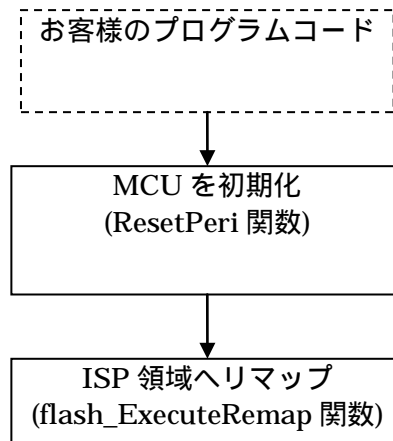
3. 送信待ち

以下の送信ウィンドウを表示し、XMODEM プロトコルでのファイル送信待ちになります。
相手(ターゲットボード)から送信要求が来ると送信を開始し、送信が終了すると送信ウィンドウは自動的に閉じます。



3. お客様のプログラムから ISP プログラムを起動する手順

お客様のプログラムコードから、ISP プログラムへの切り替え手順を以下に示します。



ResetPeri 関数は、サンプルとしてご提供しております。使用したペリフェラルを初期状態に戻す処理を追加してください。

改版履歷

改版履歴

版数	発行日	ページ		変更内容
		改版前	改版後	
1	2014.4.21	-	-	初版発行