

# HTU8

## ユーザーズマニュアル

---

プログラム開発支援ソフトウェア

## 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。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。また、その他の製品名や社名などは、一般に商標または登録商標です。

Copyright 2009 - 2011 LAPIS Semiconductor Co., Ltd.

---

# ラピスセミコンダクタ株式会社

〒193-8550 東京都八王子市東浅川町 550 番地 1

<http://www.lapis-semi.com/jp/>

## 目次

1. はじめに.....	1
2. ROMコードデータについて .....	2
2.1 開発完了HEXデータとROMコードデータ .....	2
2.2 ROMコードデータ.....	3
2.3 プログラム開発時点での推奨事項.....	4
3. ラピスセミコンダクタにご提供頂くファイルと情報 .....	5
3.1 HTU8 の入出力概要.....	6
4. ROMコードデータの作成手順.....	7
4.1 ABSファイルから「開発完了HEXデータ」に変換する.....	7
4.2 HTU8 を使用し「開発完了HEXデータ」から「ROMコードデータ」を作成する.....	9
4.2.1 U8 用のコマンドプロンプトを開く .....	9
4.2.2 「開発完了HEXデータ」が格納されているフォルダに移動する.....	9
4.2.3 「開発完了HEXデータ」から「ROMコードデータ」を生成する.....	10
4.3 「ROMコードデータ」を使用して最終評価を行う(フラッシュROMマイコンの場合のみ) ....	12
4.4 「ROMコードデータ」をラピスセミコンダクタに送る .....	13
5. HTU8 の説明 .....	14
5.1 コマンドラインの書式 .....	14
5.2 入出力ファイル .....	14
5.2.1 入力ファイル.....	14
5.2.2 出力ファイル.....	15
5.3 オプション .....	17
5.4 応答ファイル .....	18
5.5 HTU8 が出力するメッセージ.....	19
5.5.1 起動メッセージ.....	19
5.5.2 終了メッセージ.....	20
5.6 終了コード .....	20
5.7 エラーメッセージ.....	21
5.8 制限事項 .....	24



## 1. はじめに

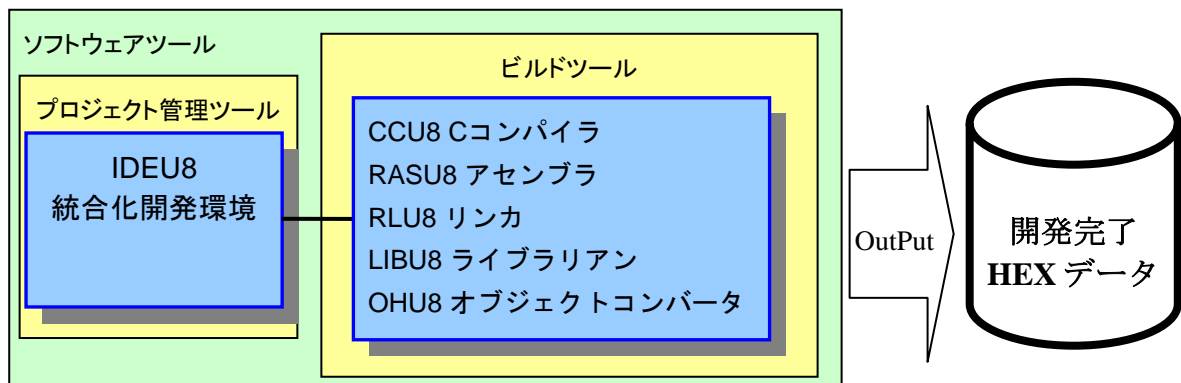
本マニュアルは、nX-U8 コアをベースとするマイクロコントローラ（以降、U8 マイコンと呼びます）のマスク ROM またはフラッシュ ROM に、お客様のプログラムデータをラピスセミコンダクタ工場にて書き込むためのデータ（以降、ROM コードデータと呼びます）を作成する方法について、説明しています。

お客様がラピスセミコンダクタ工場にお客様の ROM コードデータの書き込み依頼をする場合には、必ず、本マニュアルをお読みください。

## 2. ROM コードデータについて

### 2.1 開発完了 HEX データと ROM コードデータ

開発完了HEXデータとは、お客様が「図 2-1 開発時に使用するツール」を使用して作成したデータのことです。



※上記ツールについては、各ツールのユーザーズマニュアルを参照してください。

図 2-1 開発時に使用するツール

ROMコードデータとは、「図 2-2 ROMコードデータ作成ツール」を使用して作成したデータのものです。このデータをラピスセミコンダクタ工場にて書き込みます。

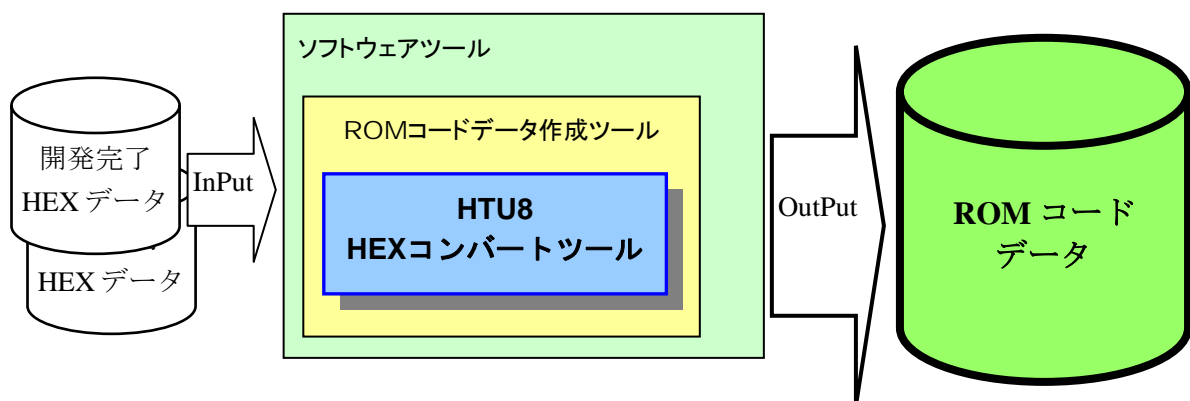


図 2-2 ROM コードデータ作成ツール

## 2.2 ROM コードデータ

マスク ROM またはフラッシュ ROM にデータを書き込む際、データが定義されていない領域（以降、ブランク領域と呼びます）には、ブランク領域に 0FFFFH(\*注)を埋め込んでおくことを推奨しています。

ブランク領域の内容は不定であり、問題が発生して U8 マイコンがブランク領域を実行した場合、U8 マイコンは予測できない動作となるためです。

HTU8 を使用することによって、開発完了 HEX データに対し、ブランク領域に 0FFFFH を埋めた ROM コードデータを作成することができます。

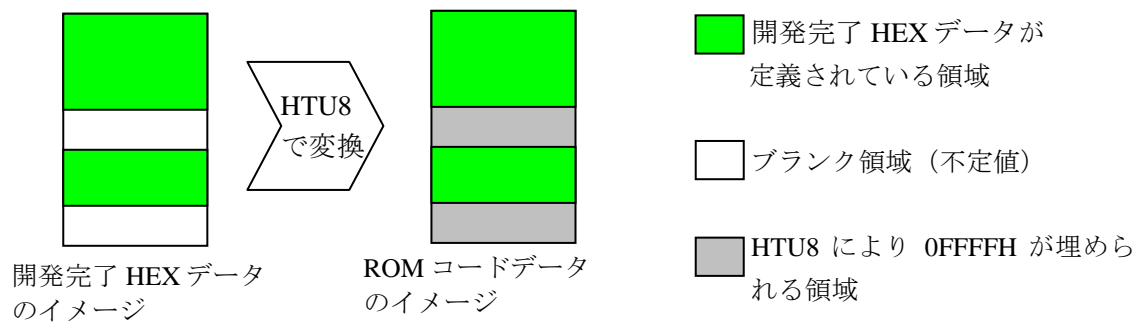


図 2-3 ROM コードデータのイメージ

(\*注) 0FFFFH は、U8 マイコンのブレーク命令コードです。

## 2.3 プログラム開発時点での推奨事項

ROM コードデータのトラブルを防ぐために、プログラム開発時に以下のことを推奨します。

- (1) ブレーク命令が発生した場合の処理を記述することを推奨します。

記述方法については、『CCU8 プログラミングガイド』の「2.5.6 リセット処理ルーチンの記述」を参照してください。

- (2) IDEU8 でのオプション設定において、ビルド時に開発完了 HEX データを毎回作成するように設定することを推奨します。

設定方法については、「4.1 ABS ファイルから「開発完了 HEX データ」に変換する」を参照してください。

- (3) DTU8 デバッガにてデバッグをする際には、空き領域を 0FFH で埋めるように設定することを推奨します。

DTU8 デバッガでプログラムファイルを読み込む際に空き領域を 0FFH で埋めておくことで、デバッグ時に HTU8 が生成する ROM コードデータと同じ状態にすることができます。設定方法については、『DTU8 ユーザーズマニュアル』の「3.4.1 プログラムファイルの読み込みダイアログボックス」を参照してください。

- (4) フラッシュ ROM マイコンの場合には、ROM コードデータをフラッシュ ROM に書き込み、お客様のシステム上で最終評価を行うことを推奨します。

方法については、『4.3 「ROMコードデータ」を使用して最終評価を行う』を参照してください。



### 3. ラピスセミコンダクタにご提供頂くファイルと情報

お客様がラピスセミコンダクタにご提供頂くファイルと情報を以下に示します。なお、これらは、HTU8 で作成します。

#### (1) ご提供頂くファイル

No.	分類	HTU8 で作成されるファイル名	説明	備考
1	ROM コードデータファイル	<i>TargetName_nnnRA.hex</i> または <i>TargetName_nnnRA.s</i>	ブランク領域を 0FFFFH で埋めた ROM コードデータファイル。	2つのファイルの内容は、同一です。ラピスセミコンダクタにてこれらのファイルを受領時に、ファイル内容が一致していることを確認することで、データが壊れていないことを確認します。
2	(同上)	<i>targetnamt_nnnRB.hex</i> または <i>TargetName_nnnRB.s</i>	ブランク領域を 0FFFFH で埋めた ROM コードデータファイル。	
3	ログファイル	<i>TargetName_nnnRA.log</i>	変換作業に関する情報が含まれたテキストファイル	この中にはチェックサム情報が含まれています。

*TargetName\_nnn* は HTU8 の出力ファイルのベース名を示します。

#### 出力ファイル名について

お客様が HTU8 を使用して上記のファイルを作成する際は、以下のルールに従って出力ファイルの名前を指定してください。

##### “*TargetName\_nnn*”

*TargetName* は対象のマイクロコントローラの名前です。ML610Q340、ML610Q431 などご使用のマイクロコントローラの名前を指定してください。

*nnn* は、ラピスセミコンダクタから通知される 3 桁の番号（以降、ROM コード番号と呼びます）です。

*TargetName* と *nnn* の間にはアンダスコア ( \_ ) を入れてください。

例えば、マイクロコントローラ名が ML610Q431、ROM コード番号が 010 の場合、HTU8 で指定する出力ファイルのベース名は、ML610Q431\_010 となります。

#### (2) ご提供頂く情報

No.	分類	説明	備考
1	チェックサム	作成した「フラッシュ ROM コードデータ」のチェックサム	このチェックサムは、上記のログファイルに出力されています。

### 3.1 HTU8 の入出力概要

HTU8 の入出力概要を以下に示します。

以下の図に示すように、HTU8 は開発完了 HEX データ（拡張 Intel HEX ファイル、または、モトローラ S2 ファイル、または、モトローラ S3 ファイル）を入力とし、ブランク領域を 0FFFFH で埋めた 2 つの ROM コードデータファイル（拡張 Intel HEX ファイルまたは、モトローラ S2 フォーマット）とチェックサム情報が含まれた 1 つのログファイルを出力します。

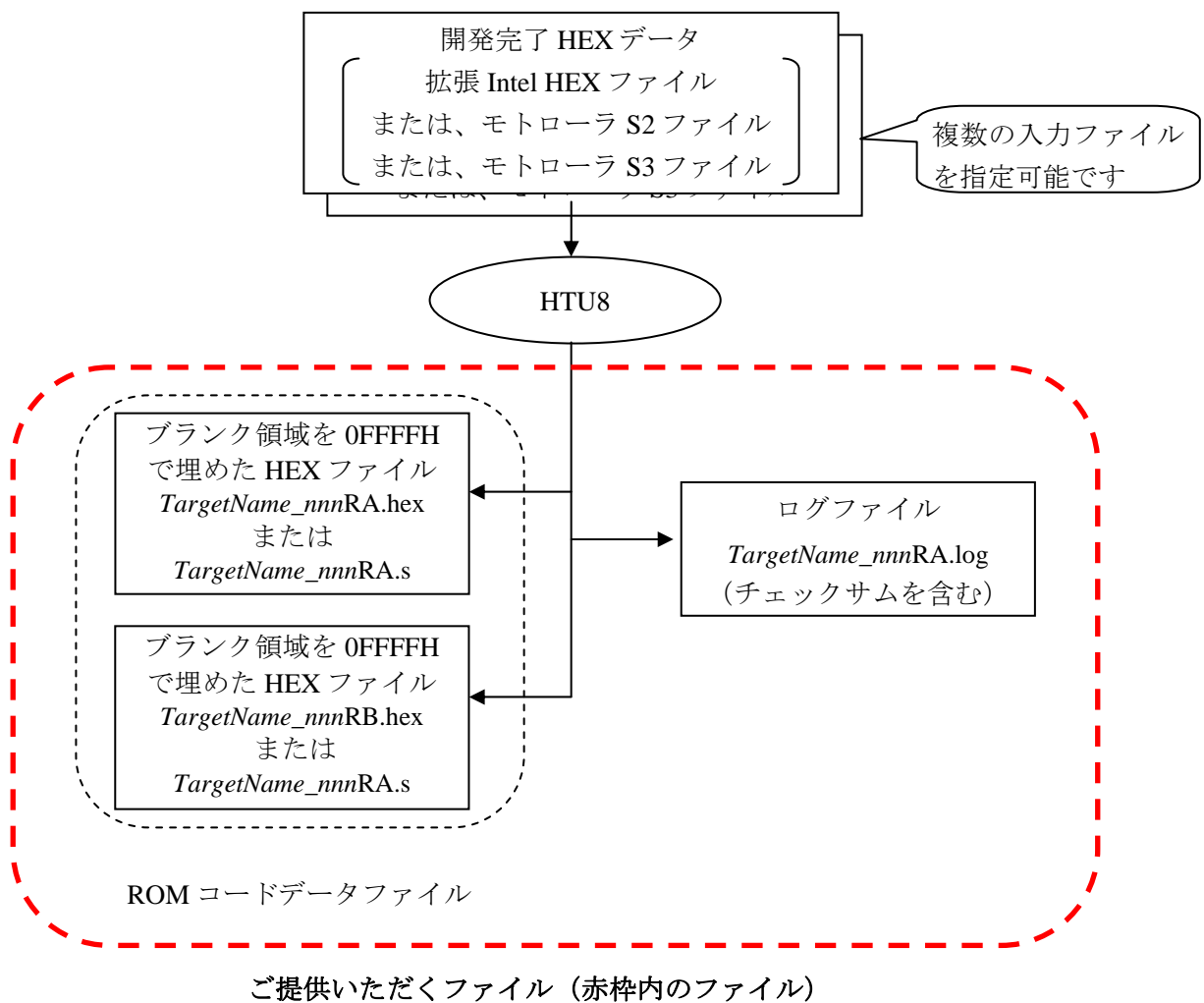


図 3-1 HTU8 の入出力概要

次章以降で ROM コードデータの作成手順を説明します。

## 4. ROM コードデータの作成手順

ROMコードデータを作成するためには、「図 2-1 開発時に使用するツール」とHTU8 が必要です。

以下に手順を示します。

### 4.1 ABS ファイルから「開発完了 HEX データ」に変換する

デバッグ情報を含んだアブソリュートオブジェクトファイル（以降、ABS ファイルと呼びます）を「開発完了 HEX データ」に変換します。

以下に、「開発完了 HEX データ」を作成する方法を示します。

なお、ここで説明する内容は、プログラム開発時の段階で設定しておくことを推奨します。

図 4-1に示すように、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。

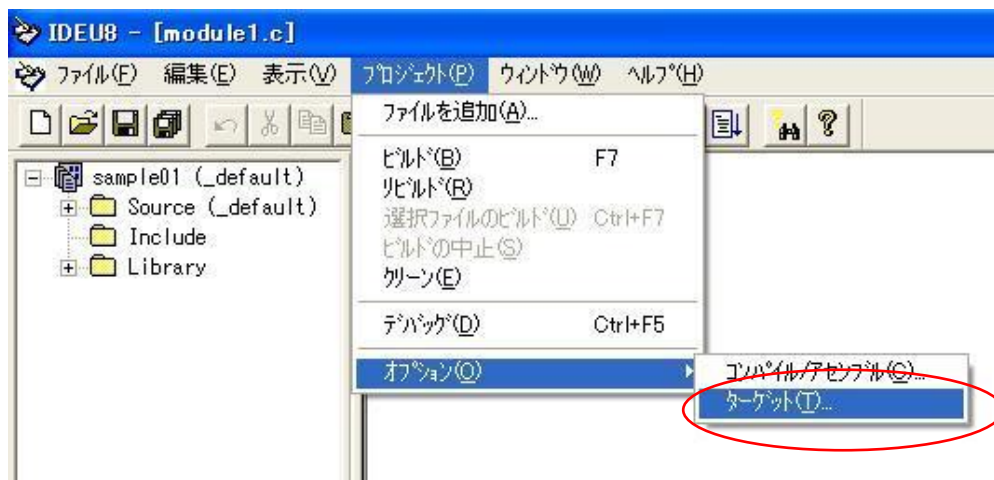


図 4-1 設定ダイアログの選択

図 4-2に示すように、[ターゲット オプション]ダイアログが表示されますので、[一般]タブを選択します。

[オブジェクトコンバータ]グループの[HEX 形式ファイルを出力する]をチェックし、[インテル HEX]を選択します。[デバッグ情報を出力する]のチェックは、外してください。

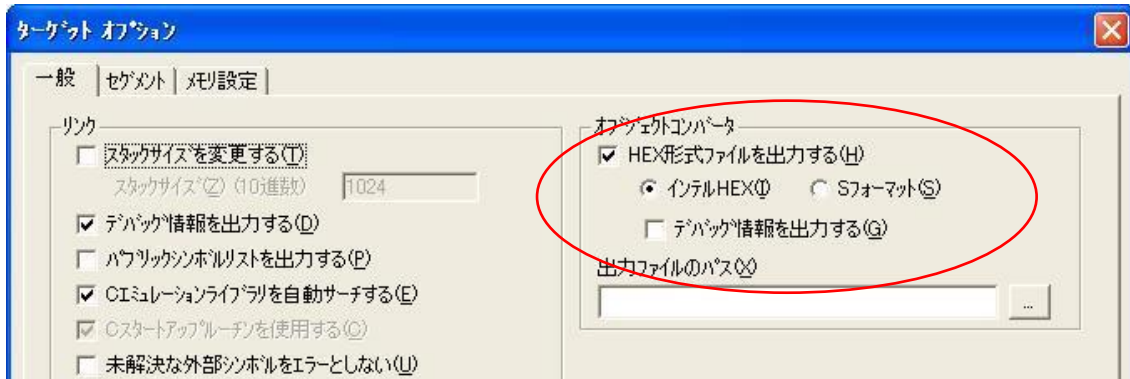


図 4-2 ターゲット オプションダイアログ

その他必要な設定をした上でプログラムをビルドします。プログラムのビルド後、図 4-3に示すように、アウトプットウィンドウに“Convert End.” “ビルド終了”と表示されていることを確認してください。これで、指定されたフォルダに「開発完了HEXデータ」のファイルが生成されます。

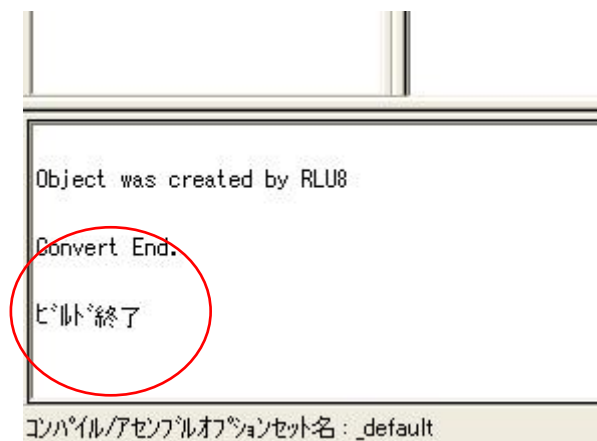


図 4-3 変換終了メッセージ

## 4.2 HTU8 を使用し「開発完了 HEX データ」から「ROM コードデータ」を作成する

### 4.2.1 U8 用のコマンドプロンプトを開く

[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 用コマンドプロンプトを開きます。

### 4.2.2 「開発完了 HEX データ」が格納されているフォルダに移動する

U8 用コマンドプロンプト上で、「CD コマンド」を用いて変換対象の「開発完了 HEX データ」が格納されているフォルダに移動します。

例えば、変換対象の「開発完了 HEX データ」が格納されているフォルダが”C:¥Test¥Sample¥Hex”であれば、次のように入力します。

例

```
CD C:¥Test¥Sample¥Hex
```

### 4.2.3 「開発完了 HEX データ」から「ROM コードデータ」を生成する

U8 用コマンドプロンプト上で、HTU8 を起動します。

以下に 2 つの例を示します。

#### 例 1

```
HTU8 Test01.hex /TM610431 /FML610Q431_010 /OH
```

上の例では、Test01.hex を入力ファイルとして、ROM コードデータを生成しています。

/T オプションは対象の機種毎に用意された初期化用ファイルを指定するためのオプション、/F オプションは出力ファイル名を指定するためのオプション、/OH は出力ファイルのフォーマットを指定するオプションです。

各オプションの詳細については、「5.3 オプション」を参照してください。

上記のように起動した場合、正常に処理が終了すると、ML610Q431\_010RA.hex と ML610Q431\_010RB.hex という 2 つの ROM コードデータファイルと、ML610Q431\_010RA.log というログファイルが出力されます。

#### 例 2

```
HTU8 @Sample.res
```

上の例は、応答ファイルを使用した例です。応答ファイルは、入力ファイルやオプションを記述したテキストファイルです。複数のファイルを入力とする場合や、入力ファイル中の一部のデータを読み込んで指定したアドレスに割り付けたい場合には、応答ファイルを使用すると便利です。

応答ファイルの記述例を以下に示します。

```
// 応答ファイル記述例 (Sample.res)
Test01.hex // 入力ファイル①
Data01.s /AL(0, 7FFFH, 1:0000H) // 入力ファイル②
Data01.s /AL(8000H, 0FFFFH, 4000H) // 入力ファイル③
/TM610340 // 機種指定
/FML610Q340_020 // 出力ファイル名
/OS // 出力フォーマット指定
```

上の例では、3 つの入力ファイルをマージして ROM コードデータを生成しています。入力ファイル①に対しては読み込んだ内容をそのまま ROM コードデータファイルに出力し、入力ファイル②、③に対しては/AL オプションを使用して、ファイル中の一部のデータを読み込んで指定したアドレスに割り付けるように指示しています。

各オプションの詳細については、「5.3 オプション」を参照してください。

上記のように起動した場合、正常に処理が終了すると、ML610Q340\_020RA.s と ML610Q340\_020RB.s という 2 つの ROM コードデータファイルと、ML610Q340\_020RA.log というログファイルが出力されます。

入力ファイルと ROM コードデータの関係は次のようになります。

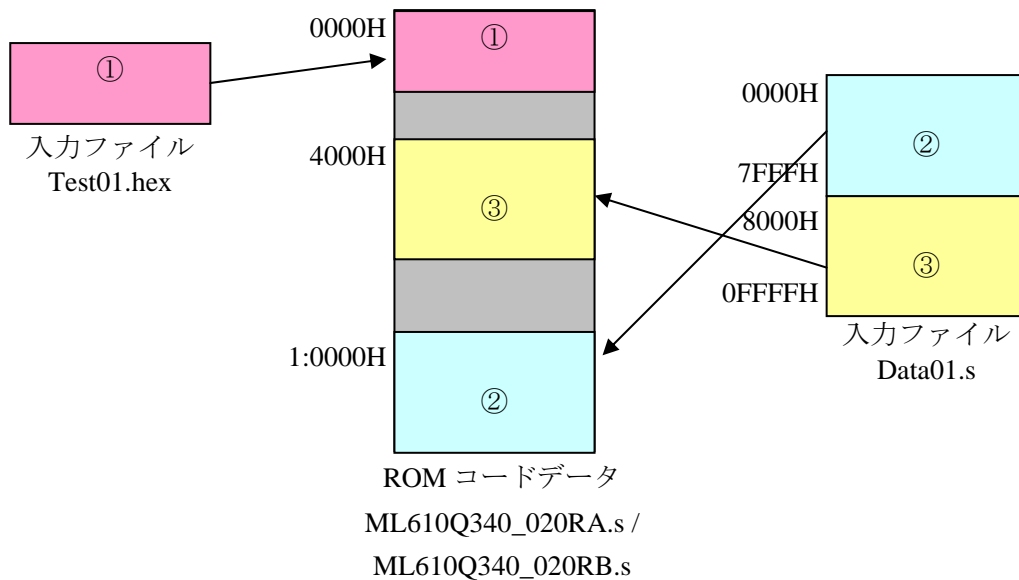


図 4-4 入力ファイルと ROM コードデータの関係

正常に処理が終了すると下記のような終了メッセージを表示します。

```
*** The Result of HTU8 ***
Status          : OK
Total checksum   : XXXXXXXXXXXX
Target Name      : target
```

XXXXXXXXXX は、作成した「ROM コードデータ」のチェックサムです。

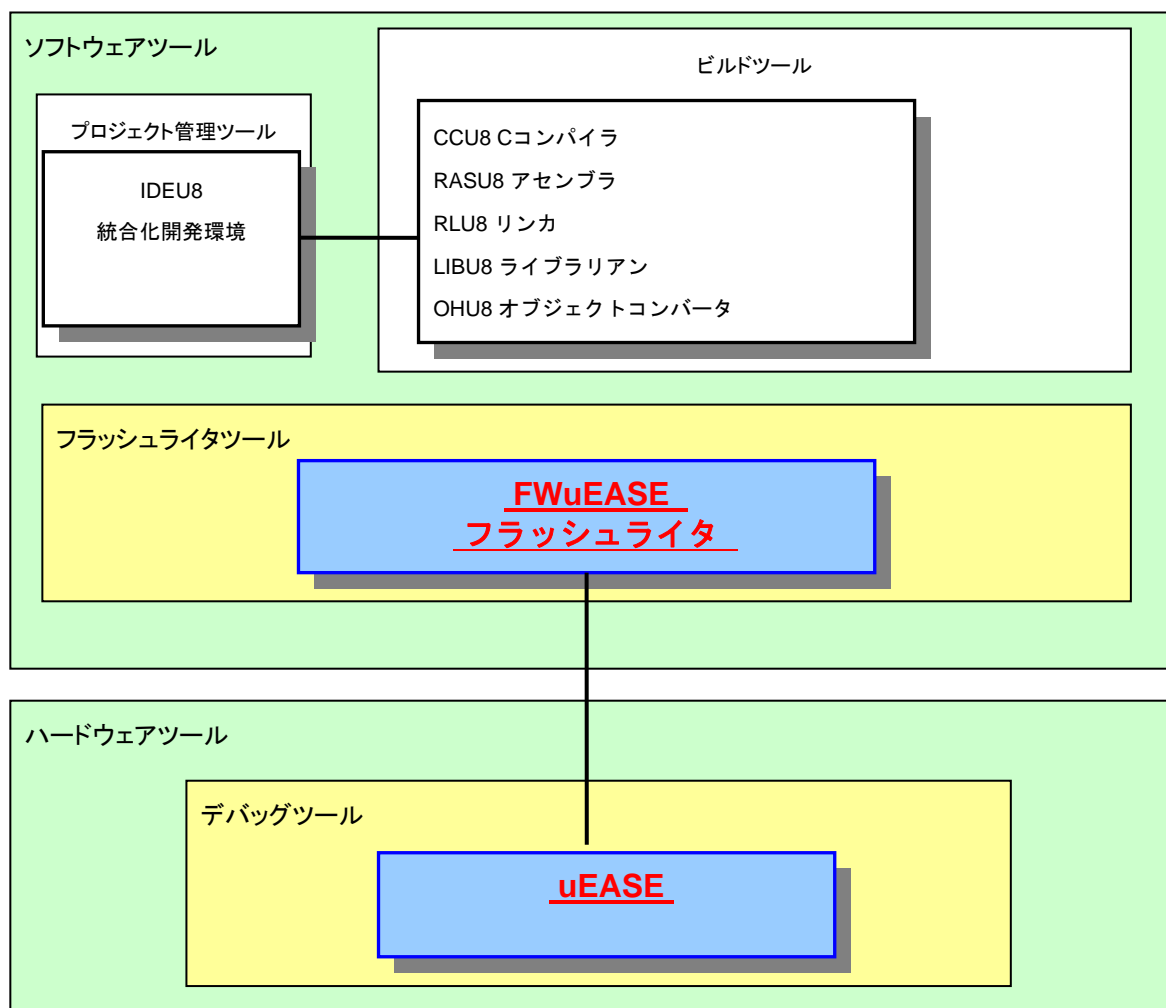
target は、/T オプションで指定された機種名です。

上記メッセージの Status が OK であれば、問題なく変換処理が行われたことになります。

### 4.3 「ROM コードデータ」を使用して最終評価を行う（フラッシュ ROM マイコンの場合のみ）

フラッシュ ROM マイコンの場合、作成した ROM コードデータをフラッシュ ROM に書き込み、お客様のシステム上で最終評価を実施してください。

「ROMコードデータ」をフラッシュROMに書き込む場合は、「図 4-5 最終評価時に使用するツール」のハードウェアツールのuEASEとソフトウェアツールのFWuEASE フラッシュライタを使用します。



※上記ツールについては、各ツールのユーザーズマニュアルを参照してください。

図 4-5 最終評価時に使用するツール



## 4.4 「ROM コードデータ」をラピスセミコンダクタに送る

ROM コードデータを使用した最終評価完了後、最終評価で使した「ROM コードデータ」のファイル（*TargetName\_nnnRA.hex* と *TargetName\_nnnRB.hex*、または *TargetName\_nnnRA.s* と *TargetName\_nnnRB.s*）とログファイル（*TargetName\_nnnRA.log*）をラピスセミコンダクタまでお送りください。また、ログファイルに出力されているチェックサムも合わせてお知らせください。

## 5. HTU8 の説明

### 5.1 コマンドラインの書式

HTU8 のコマンドラインの書式は、以下のとおりです。

*HTU8 inputfiles options*

*HTU8 @response\_file*

*inputfiles*には、「開発完了HEXデータ」のファイル名を指定します。入力ファイルは複数指定することができます。指定可能な入力ファイルについては、「5.2.1 入力ファイル」を参照してください。

*options*には、オプションを指定します。オプションの指定位置は任意です。オプションの詳細については、「5.3 オプション」を参照してください。

@マークに続く *response\_file* は応答ファイルとして扱い、指定されたファイル中に記述された内容を入力として読み込みます。応答ファイルについては「5.4 応答ファイル」を参照してください。

### 5.2 入出力ファイル

#### 5.2.1 入力ファイル

HTU8 の入力ファイルとして扱えるファイルは、拡張 Intel HEX フォーマットのファイル、または、モトローラ S2 フォーマット、または、モトローラ S3 フォーマットのファイルです。

入力ファイルは複数指定できます。指定できる数は最大 16 個までです。

入力ファイルには、デバッグ情報を含めないようにしてください。デバッグ情報を含めないファイルの作成方法については、『4.1 ABSファイルから「開発完了HEXデータ」に変換する』を参照してください。

デバッグ情報を含むファイルを入力ファイルとして指定した場合には、HTU8 はエラーを出力します。

## 5.2.2 出力ファイル

HTU8 が出力するファイルは、内容が同じで名前の異なる 2 つのフラッシュ ROM コードデータファイルと、変換の結果を記録したログファイルです。

出力ファイルの名前は、/F オプションで指定された出力ファイルのベース名と出力ファイルフォーマット指定のオプションに基づいて HTU8 が付与します。

出力ファイルのベース名として指定できる文字数は最大で 30 文字までです。

出力ファイルの名前の命名則は以下のとおりです。

出力ファイル ベース名	出力フォーマット 指定オプション	ROM コードデータファイル名		ログファイル名
		出力ファイル A	出力ファイル B	
<i>TargetName_nnn</i>	/OH	<i>TargetName_nnnRA.hex</i>	<i>TargetName_nnnRB.hex</i>	<i>TargetName_nnnRA.log</i>
	/OS	<i>TargetName_nnnRA.s</i>	<i>TargetName_nnnRB.s</i>	<i>TargetName_nnnRA.log</i>

*TargetName\_nnn* は出力ファイルのベース名を示します。

### 5.2.2.1 ROM コードデータファイル

上記の表の出力ファイル A と出力ファイル B に該当します。出力ファイル A と出力ファイル B の内容はいずれも同じです。

入力ファイルのデータが存在しないブランク領域は 0FFFFH で埋められたものとなります。

### 5.2.2.2 ログファイル

ログファイルの出力形式は以下のとおりです。

#### (1) 処理が正常に終了した場合

※斜体は、HTU8 実行条件（オプションなど）により異なります。

ログファイルの表示内容	説明
HTU8, Ver.1.20	①HTU8 バージョン
Copyright (C) 2011 LAPIS Semiconductor Co., Ltd.	-
Tue Feb 12 23:11:31 2009	②HTU8 を実行したタイムスタンプ
*** The Result of HTU8 ***	-
Status : OK	③HTU8 の実行結果
Total checksum : 0000FE5A8D	④チェックサム値 チェックサムの桁数は、10 桁です。 10 桁に満たなかった場合は、「0」が 付加されます。
Target Name : M610340	⑤/T オプションで指定された機種名
*** COMMAND LINE ***	⑥コマンドライン入力の内容
test1.hex /AL(1000H, 3FFFH, 1:0000H)	-
/FML610Q340_020 /M610340 /OS	⑦入力ファイル名
*** INPUT FILES ***	⑧入力ファイルのフォーマット
File Name : TEST1.HEX	⑨入力ファイルに記載されている開始ア ドレスと終了アドレス
Format : Extended Intel Hex	⑩入力ファイルから読み込むデータのアド レス範囲
Address Range : 00H:0000H 00H:5FFFH	⑪配置先アドレス
>Allocation Range : 00H:1000H 00H:3FFFH	-
Destination : 01H:0000H	⑫出力ファイル名
*** OUTPUT FILES ***	⑬出力ファイルのフォーマット
File Name : ML610Q340_020RA.s	⑭出力ファイルに記載されている開始ア ドレスと終了アドレス
ML610Q340_020RB.s	⑮ログファイルの終了
Format : Motorola S	
Address Range : 00H:0000 00H:0FFFFH	
01H:0000 01H:07FFFH	
<End of Log>	

#### (2) 異常終了した場合

※斜体は、HTU8 実行条件（オプションなど）により異なります

ログファイルの表示内容	説明
HTU8, Ver.1.20	①HTU8 バージョン
Copyright (C) 2011 LAPIS Semiconductor Co., Ltd.	-
Tue Feb 12 23:11:31 2009	②HTU8 を実行したタイムスタンプ
*** The Result of HTU8 ***	-
Status : NG	③HTU8 の実行結果
Error No : xxx	④エラー番号

## 5.3 オプション

HTU8 のオプションを以下に示します。

### /OH

出力ファイルのフォーマットを拡張 Intel HEX フォーマットに指定します。

出力できるデータの範囲は 0~0FH:0FFFFH までです。

### /OS

出力ファイルのフォーマットをモトローラ S2 フォーマットに指定します。

出力できるデータの範囲は 0~0FH:0FFFFH までです。

/OH と /OS はどちらか 1 つのみ指定可能です。/OH、/OS のどちらも省略した場合は、/OH が指定されたものとして扱います。

### /Ttarget

対象となる機種を指定します。*target* には、マイクロコントローラ名から 'L' と 'Q' を取り除いた名前を指定してください。ML610Q431 の場合、/TM610431 と指定します。

HTU8 は /T オプションで指定された *target* に拡張子 ".tmi" を付けたファイル (TMI ファイルと呼びます) を検索し、その TMI ファイルからメモリ領域の情報を取得します。そして、そのメモリ領域の全域を出力範囲とします。

TMI ファイルは、通常は U8 開発環境をインストールしたフォルダの下の Tmi フォルダに格納されています。

### /R(start\_addr, end\_addr)

出力範囲を指定します。

通常は /R オプションを指定する必要はありません。機種に対応する TMI ファイルが存在しない場合に指定するオプションです。

/T オプションと /R オプションはどちらか一方を必ず指定する必要があります。

**/Ppath\_name**

TMI ファイルを検索するパスを指定します。

通常はこのオプションを指定することはありません。TMI ファイルを見つけることができなかった場合に検索するパスを指定するためのオプションです。

**/Foutput\_name**

出力ファイルのベース名を *output\_name* として指定します。*output\_name* にはパスを含めて指定できます。パスを指定しなかった場合、出力ファイルおよびログファイルはカレントディレクトリに格納されます。パスを指定した場合、出力ファイルおよびログファイルは指定されたディレクトリに格納されます。

本オプションを指定しなかった場合は、先頭に指定された入力ファイルの名前を出力ファイルのベース名とします。

実際に出力されるファイルの名前については、「5.2.2 出力ファイル」を参照してください。

**/AL(start\_addr, end\_addr, allocation)**

直前に指定された入力ファイル中の *start\_addr* – *end\_addr* の範囲にあるデータを読み込んで、*allocation* で指定されたアドレスに配置します。

*start\_addr*, *end\_addr*, *allocation* には、10 進数、16 進数の指定が可能です。数値の先頭に 0x または末尾に H を記述した場合には 16 進数が指定されたものとみなします。それ以外の場合には 10 進数が指定されたものとみなします。

物理セグメントとオフセットはコロン ':' で区切って指定してください。物理セグメントおよびコロンを省略した場合は物理セグメント 0 が指定されたものとみなします。

```
Data01.s /AL(0, 7FFFH, 1:0000H)
```

上記のように指定した場合、入力ファイル *Data01.s* の 0-7FFFH の範囲のデータを読み込んで、1:0000H に配置します。

一つの入力ファイルに対し指定できる /AL オプションは一つのみです。一つのファイルから複数の範囲のデータを読み込んでそれぞれを異なるアドレスに配置する場合には、次のように分けて指定してください。

```
Data01.s /AL(0, 7FFFH, 1:0000H) Data01.s /AL(8000H, 0FFFFH, 4000H)
```

## 5.4 応答ファイル

応答ファイルは、入力ファイルやオプションを記述したテキストファイルです。複数のファイルを入力とする場合や、入力ファイル中の一部のデータを読み込んで指定したアドレスに割り付けたい場合には、応答ファイルを使用すると便利です。

応答ファイルにはコマンドラインの入力内容を複数の行に分けて記述することができます。

また、コメントを記述することもできます。コメントは “#” または “//” で始まり、行末までとなります。

指定できる応答ファイルは 1 つだけです。応答ファイルの中に応答ファイルを指定すること

はできません。

以下に応答ファイルの記述例を示します。

```
#####
# 応答ファイル記述例 (Sample.res)
# `#`または`///`から行末まではコメントとみなします
#####
Test01.hex                // 入力ファイル 1
Data01.s /AL(0, 7FFFH, 1:0000H) // 入力ファイル 2
Data01.s /AL(8000H, 0FFFFH, 4000H) // 入力ファイル 3
/TM610340                // 機種指定
/FML610Q340_020          // 出力ファイル名
/OS                       // 出力フォーマット指定
```

上記の応答ファイルを使用して HTU8 を起動する場合は、以下のように指定します。

```
HTU8 @Sample.res
```

応答ファイルにコマンドラインの一部を記述することもできます。

```
// 応答ファイル記述例 (Sample2.res)
Data01.s /AL(0, 7FFFH, 1:0000H) // 入力ファイル 2
Data01.s /AL(8000H, 0FFFFH, 4000H) // 入力ファイル 3
/OS // 出力フォーマット指定
```

次のように指定して HTU8 を起動した場合、

```
HTU8 Test01.hex @Sample2.res /TM610340 /FML610Q340_020
```

上記の HTU8 @Sample.res と同じ動作となります。

## 5.5 HTU8 が出力するメッセージ

HTU8 が出力するメッセージには、HTU8 の起動または終了を表すメッセージと、処理の最中に何らかの異常があったことを表すエラーメッセージがあります。

### 5.5.1 起動メッセージ

HTU8 は起動直後に、次のメッセージを画面に表示します。

```
HTU8 HEX Converter, Ver.1.20
Copyright (C) 2009-2011 LAPIS Semiconductor Co., Ltd.
```

### 5.5.2 終了メッセージ

正常終了時には、次のメッセージを表示します。

```
*** The Result of HTU8 ***  
Status           : OK  
Total checksum   : XXXXXXXXXXXX
```

XXXXXXXXXXXX は、作成した HEX ファイルのチェックサムです。

エラー発生時には、次のメッセージを表示します。

```
Error nnn : error_message
```

*nnn* はエラー番号、*error\_message* はエラーの内容を示すメッセージです。エラーメッセージについては、「5.7 エラーメッセージ」を参照してください。

### 5.6 終了コード

HTU8 は終了コードを返します。正常に終了した場合は 0、エラーにより終了した場合は 3 を返します。それ以外の値は返しません。

終了コード	終了時の状態	意味
0	正常終了	正常に終了しました。
3	エラー	実行中に問題が発生し、処理を続行できませんでした。



## 5.7 エラーメッセージ

HTU8 が出力するエラーメッセージを以下に示します。

エラー 番号	メッセージ	説明
000	Ok	正常終了
102	insufficient memory	メモリが不足しています。
153	File name too long (255 characters)	起動時に指定されたディレクトリを含むファイル名の長さが 255 文字を超えています。
154	Unable to open Input_file	起動時に指定された Input ファイルがオープンできません。(Input ファイル名に全角、または、空白を指定した場合にもこのエラーが出力されます。)
158	Bad syntax on command line	コマンドラインの指定が間違っています。
159	Invalid conversion range	/R(start_addr, end_addr)オプション指定時の範囲指定が不正です。
160	/Ttarget option or /R option is not specified	/Ttarget または /R (start_addr, end_addr)のどちらか一方は必ず指定する必要があります。
162	Path name too long (255 characters)	/P オプション指定時に指定されたパス名の長さが 255 文字を超えています。
163	Invalid Path name	/P オプション指定時に指定されたパス名が不正です。
164	/R (start_addr, end_addr) too much	/R(start_addr, end_addr)オプションの指定が、32 組を超えました。
166	Address overlap	/R(start_addr, end_addr)オプションで指定した領域がオーバーラップしています。
168	Number of input files exceed maximum (16 files)	入力ファイル数が 16 を超えています。
170	Output file name not given.	/F オプションに対し出力ファイル名が指定されていません。
171	Output file name too long (255 characters)	パスを含む出力ファイル名の長さ(“RA”を付加した長さ)が 255 文字を超えました。

エラー 番号	メッセージ	説明
172	Output file name base too long (32 characters)	出力ファイルのベース名の長さ(“RA”を付加した長さ)が 32 文字を超えました。
173	Invalid output file name	/F オプションで指定された出力ファイル名が不正です。
174	Response file name too long (255 characters)	@に続いて指定された応答ファイル名の長さが 255 文字を超えました。
175	Invalid response file name	@に続いて指定された応答ファイル名が不正です。
176	Bad syntax on command line	@が複数指定されています。指定できるのは一つだけです。
177	Unable to open response file	応答ファイルがオープンできません。
178	No input files before a /AL option.	/AL オプションの前に入力ファイルが指定されていません。
179	Bad syntax on command line	/AL オプションが 1 つの入力ファイルに対し複数指定されました。
180	Input files not given.	入力ファイルが指定されていません。
181	XNV/SNV files are not supporting.	NVRAM コードを含むファイル（拡張子が、「XNV」または、「SNV」のファイル）は指定できません。
182	Command line buffer overflow.	コマンドラインバッファをオーバーフローしました。コマンドラインバッファは 32K バイトです。
201	file not found : TMI file	/Ttarget で指定された target.tmi が見つかりません。
207	Unable to close TMI file	TMI ファイルのクローズに失敗しました。
302	Unable to read Extended Intel HEX	入力ファイル（拡張 Intel HEX ファイル）の読み込みに失敗しました。
305	Unable to read Motorola S	入力ファイル（モトローラ S フォーマットファイル）の読み込みに失敗しました。
309	Invalid Extended Intel HEX format.	入力ファイルが有効な拡張 Intel HEX ファイルではありません。
310	Invalid Motorola S format.	入力ファイルが有効なモトローラ S フォーマットファイルではありません。
311	Invalid Extended Intel HEX format.	入力ファイルに終了レコードがありません。OHU8 が出力したファイルではない可能性があります。

エラー 番号	メッセージ	説明
312	Invalid Motorola S format.	入力ファイルに開始レコードまたは終了レコードがありません。OHU8 が出力したファイルではない可能性があります。
313	Checksum failure	Checksum エラーです。入力ファイルが壊れている可能性があります。
314	Unable to open Input Files	入力ファイルのオープンに失敗しました。
315	Unable to close Input Files	入力ファイルのクローズに失敗しました。
351	Unable to open Extended Intel HEX	出力ファイル（拡張 Intel HEX ファイル）のオープンに失敗しました。
352	Unable to read Extended Intel HEX	出力ファイル（拡張 Intel HEX ファイル）の読み込みに失敗しました。
353	Unable to open Motorola S	出力ファイル（モトローラ S フォーマットファイル）のオープンに失敗しました。
354	Unable to read Motorola S	出力ファイル（モトローラ S フォーマットファイル）の読み込みに失敗しました。
355	Unable to write to Extended Intel HEX	出力ファイル（拡張 Intel HEX ファイル）への書き込みに失敗しました。
356	Unable to write to Motorola S	出力ファイル（モトローラ S フォーマットファイル）への書き込みに失敗しました。
357	Unable to close Extended Intel HEX	出力ファイル（拡張 Intel HEX ファイル）のクローズに失敗しました。
358	Unable to close Motorola S	出力ファイル（モトローラ S フォーマットファイル）のクローズに失敗しました。
359	Unable to open *.LOG	ログファイルのオープンに失敗しました。
360	Unable to write to *.LOG	ログファイルへの書き込みに失敗しました。
361	Unable to close *.LOG	ログファイルのクローズに失敗しました。
371	Out of Address Range	入力ファイルのアドレスが、出力ファイルのアドレス範囲を超えています。ターゲットデバイスが異なっている可能性があります。
372	Duplicate Address Detection.	アドレスが重複したデータが存在します。
9xx	Internal_Error_9xx	内部エラーを検出しました。9xx は内部エラーの番号です。このエラーが表示された場合には、エラーが発生したときの状況をラピスセミコンダクタまでお知らせください。

## 5.8 制限事項

HTU8 には以下の制限事項があります。

- ・ HTU8 を複数のコマンドプロンプト上から同時に起動しないでください。HTU8 を同時に起動した場合の動作は保証されません。

HTU8  
ユーザーズマニュアル

---

2011 年 10 月      第 3 版発行

©2009 - 2011 LAPIS Semiconductor Co., Ltd.

---